Ministry of Science and Higher Education of the Russian Federation
Siberian Federal University

I.P. ROZHNOV
L.A. KAZAKOVTSEV
M.V. KARASEVA

GH-VNS-BASED ALGORITHMS FOR CLUSTERING PROBLEM

MONOGRAPH

Moscow
INFRA-M
2023

УДК 519.6(075.4)
ББК 22.19
Р63

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121)

Reviewers:

Stupina A.A., Doctor of Technical Sciences, Head of the Department of Digital Management Technologies (Siberian Federal University);

Masich I.S., Doctor of Technical Sciences, Associate Professor, Professor of the Department of System Analysis (Reshetnev Siberian State University of Science and Technology)

Rozhnov I.P.

GH-VNS-based algorithms for clustering problem : monograph / I.P. Rozhnov, L.A. Kazakovtsev, M.V. Karaseva. — Moscow : INFRA-M, 2023. — 162 p. : il. — (Scientific Idea).

ISBN 978-5-16-019605-3

A problem of increasing the accuracy and stability of the automatic grouping (clustering) algorithms is considered based on a new approach to developing clustering algorithms based on parametric optimization models for k-means, k-medoid, clear clustering problems based on separation of a mixture of probability distributions (with application of the classification EM-algorithm). The study proposes new search algorithms with alternating randomized neighborhoods and parallel modifications of algorithms with a greedy agglomerative heuristic procedure for large automatic grouping problems, adapted to the CUDA architecture. Moreover, the study presents a procedure for composing optimal ensembles of automatic grouping algorithms with a combined application of the genetic algorithm of the greedy heuristic method and a consistent matrix of binary partitions for practical problems. Algorithms and procedure for composing optimal ensembles are implemented to solve the problem of dividing prefabricated lots of industrial products into homogeneous lots based on the results of non-destructive tests.

The work is intended for scientists, specialists, undergraduate and postgraduate students involved in the development of cluster analysis algorithms, as well as in improving the quality of industrial products.

УДК 519.6(075.4)
ББК 22.19

ISBN 978-5-16-019605-3

НАУЧНАЯ МЫСЛЬ
СЕРИЯ ОСНОВАНА В 2008 ГОДУ

Министерство науки и высшего образования Российской Федерации
Сибирский федеральный университет

И.П. РОЖНОВ
Л.А. КАЗАКОВЦЕВ
М.В. КАРАСЕВА

GH-VNS АЛГОРИТМЫ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ

МОНОГРАФИЯ

Москва
ИНФРА-М
2023

УДК 519.6(075.4)
ББК 22.19
Р63

Работа выполнена при поддержке Министерства науки и высшего образования Российской Федерации (грант № 075-15-2022-1121)

Рецензенты:

Ступина А.А., доктор технических наук, заведующий кафедрой цифровых технологий управления Сибирского федерального университета;

Масич И.С., доктор технических наук, доцент, профессор кафедры системного анализа Сибирского государственного университета науки и технологий имени академика М.Ф. Решетнева

Рожнов И.П.

Р63 GH-VNS алгоритмы для решения задачи кластеризации : монография / И.П. Рожнов, Л.А. Казаковцев, М.В. Карасева. — Москва : ИНФРА-М, 2023. — 162 с. : ил. — (Научная мысль).

ISBN 978-5-16-019605-3

В монографии рассмотрена проблема повышения точности и устойчивости результата работы алгоритмов автоматической группировки (кластеризации) на основе нового подхода к разработке алгоритмов кластеризации, основанных на параметрических оптимизационных моделях, для задач k-средних, k-медоид, задачи четкой кластеризации на основе разделения смеси вероятностных распределений (с применением классификационного EM-алгоритма). В работе, с использованием нового подхода, предложены новые алгоритмы поиска с чередующимися рандомизированными окрестностями, а также параллельные модификации алгоритмов с жадной агломеративной эвристической процедурой для больших задач автоматической группировки, адаптированные к архитектуре CUDA. Кроме этого, предложена процедура составления оптимальных ансамблей алгоритмов автоматической группировки с комбинированным применением генетического алгоритма метода жадных эвристик и согласованной матрицы бинарных разбиений для практических задач. Алгоритмы и процедура составления оптимальных ансамблей внедрены в эксплуатацию на производстве для решения задачи разделения сборных партий промышленной продукции на однородные партии по результатам неразрушающих тестовых испытаний.

Предназначена для научных работников, специалистов, студентов и аспирантов, занимающихся вопросами разработки алгоритмов кластерного анализа, а также вопросами повышения качества промышленной продукции.

УДК 519.6(075.4)
ББК 22.19

ISBN 978-5-16-019605-3

TABLE OF CONTENTS

INTRODUCTION

Due to the accelerated growth of data volumes, the need for modern means and systems for collecting, storing and processing data arrays is also growing; as a result, their diversity is increasing. The increasing use of large-dimensional data sets stimulates an increased interest in the development and application of methods and tools for processing and analyzing these data sets. One of the promising areas is cluster analysis. It helps to group objects into homogeneous groups (clusters), and solving the problem of automatic grouping (clustering) comes down to developing an algorithm that can detect these groups without using pre-marked data.

There exist production problems of automatic objects grouping that must be solved relatively quickly, and the result must be such that it would be difficult to improve it applying the known methods without a significant increase in time.

The analysis of the existing problems concerns the application of methods for automatic objects grouping. They are the subject to high requirements for the accuracy and stability of the result, shows a lack of algorithms capable of producing results in a fixed time that would be difficult to improve by the known methods, and which would ensure the stability of the results with multiple runs of the algorithm. Moreover, the known algorithms (for example, a greedy heuristic method) require significant computational costs. The study is aimed at the development of improved algorithms for automatic grouping problems, which impose high requirements on accuracy and stability result taking into account a certain deficit of automatic grouping methods that are compromise in the quality of the result and the computation time (the quality means the accuracy, i.e., the closeness of the objective function value to the global optimum).

Currently, there exist tasks of automatic grouping (clustering) in any discipline that involves multivariate data analysis. There are many different methods and algorithms for automatic grouping. The most famous model of cluster analysis is a k-means model, which was proposed by Steinhaus (1957). At the same time, Lloyd developed and compiled the algorithm (although the work was published only in 1982). Since then, the k-means algorithm, its improvement, modification and combination with other algorithms, has become the topic of some researchers.

First of all, it is necessary to highlight B. Duran, P. Odell, I. Mandel, J. McQueen among the scientists who were developing automatic grouping of objects. Models of automatic grouping often have similarities with the models of the theory of object placement, and sometimes even identical to them; therefore, they were often considered jointly by scientists. A significant contribution to this research was made by Dresner C., Hamakher H., Brimberg D., Mladenovich N. (location problems), Vesolovsky V. (a wide range of problems), Hakimi S. (problems on a network), Lov R. (continuous problems with different metrics). In the USSR, Khachaturov V.R. and Cherenin V.P. investigated the issue of the location of enterprises. At the Institute of Mathematics named after S.L. Sobolev of SB RAS works of E.Kh. Gimadi, V.L. Beresnev, A.A. Kolokolov, and later Yu.A. Kochetov, A.V. Eremeev, G.G. Zabudsky, T.V. Levanova and others in the development of models of standardization and unification laid the foundation for the development of the software and mathematical tools for solving problems of automatic grouping and the theory of object location.

The search method with alternating neighborhoods, developed by N. Mladenovich and P. Hansen, became a popular method for solving discrete optimization problems (which is reflected in the works of Yu.A. Kochetov, F.G. Lopez, J. Brimbern, T.V. Levanova, Alekseeva E.V. and others), which allows finding good suboptimal solutions to rather large problems of automatic grouping.

Kazakovtsev L.A. and Antamoshkin A.N. proposed the application of algorithms with a greedy heuristic procedure and their advantage over the considered classical algorithms of automatic grouping (k-means, PAM, j-means, etc.) for multidimensional data is shown (2014). The method is an extended approach for constructing pseudo-Boolean optimization and clustering procedures. The greedy heuristic method uses evolutionary algorithms as one of the possible ways to organize a global search, including the approaches of the Krasnoyarsk school of evolutionary algorithms.

This work is devoted to solving the problem of developing and studying automatic grouping algorithms with the increased requirements for the accuracy and stability of the result with the combined use of search algorithms with alternating randomized neighborhoods and greedy heuristic algorithms for automatic grouping, including for massively parallel systems.

The main idea of this study is the combined application of alternating neighborhood search methods and greedy heuristics for automatic grouping problems, including the development of new algorithms for the greedy heuristic method using search algorithms with alternating randomized neighborhoods.

The object of the research is problems of automatic grouping of multidimensional data; the subject of the research is algorithms for solving these problems.

The aim of the study is to improve the efficiency of systems for automatic objects grouping. These systems are subject to high requirements for the accuracy and stability of the result (improving the achieved value of the objective function for a given time).

Tasks to be solved in the process of achieving the stated goal:

1. Analysis of the existing problems when applying the methods of automatic grouping of objects, which are the subject to high requirements for the accuracy and stability of the result.

2. Development of new search algorithms with alternating randomized neighborhoods and greedy heuristic procedures for the k-means problem.

3. Development of new search algorithms with alternating randomized neighborhoods and greedy heuristic procedures for the k-medoid problem.

4. Development of a combined algorithm based on the classification EM-algorithm (CEM - Classification Expectation Maximization) applying the search with alternating randomized neighborhoods and greedy heuristic procedures.

5. Implementation of greedy heuristic method algorithms for automatic grouping problems for massively parallel systems.

6. Development of a procedure for compiling ensembles of algorithms for automatic grouping, which makes it possible to increase the accuracy of separation (i.e., to reduce errors) of a prefabricated batch of industrial products into homogeneous batches of industrial products applying non-destructive test data.

Methods of system analysis, operations research, optimization theory, parallel computing were applied to solve the set tasks.

As a result of the research:

1. A new approach to the development of automatic grouping algorithms based on parametric optimization models with the combined use of search algorithms with alternating randomized neighborhoods and greedy agglomerative heuristic procedures is proposed. It is shown that the application of this approach makes it possible to create efficient algorithms for automatic grouping (according to the achieved value of the objective function in a fixed time), based on various optimization models.

2. New search algorithms with alternating randomized neighborhoods have been developed for k-means, k-medoid, clear clustering problems based on the separation of a mixture of probability distributions (using the classification EM-algorithm) applying a new approach. The study presents new algorithms that help obtaining a more accurate and stable result (in terms of the achieved value of the objective function) in comparison with the known algorithms for automatic grouping in a fixed time to apply algorithms in an interactive mode of decision making for practical problems.

3. Parallel modifications of algorithms with a greedy agglomerative heuristic procedure for large automatic grouping problems, adapted to the CUDA architecture, are proposed. It was found that the parallel implementation of the local search algorithm, as well as individual steps of the greedy agglomerative heuristic procedure, makes it possible to build an automatic grouping algorithm with a high acceleration factor, which reduces the computation time by tens of times without deteriorating the achieved value of the objective function.

4. A procedure for composing optimal ensembles of automatic grouping algorithms with the combined application of the genetic algorithm of the greedy heuristic method and a consistent matrix of binary partitions for practical problems is proposed. It was found that the accuracy of dividing a prefabricated batch of industrial products with special quality requirements into homogeneous lots, performed using the obtained ensembles, is higher than the average percentage of the results of separation accuracy using individual algorithms selected for compiling an ensemble on a set of test tasks.

The theoretical significance of the research results is in the development of a new approach to the creation of automatic grouping algorithms based on parametric optimization models, with the combined use of search algorithms with alternating randomized neighborhoods and greedy agglomerative heuristic procedures, developing a method of greedy heuristics, as well as procedures for composing optimal ensembles clustering algorithms.

The practical value of the new approach to solving problems of automatic grouping with increased requirements for the accuracy and stability of the result is due to a wide range of areas of their application in the tasks of cluster analysis, including directly in practical problems in production, where it is required to ensure high accuracy of dividing production batches of industrial products into homogeneous batches based on test results.

The software implementation of new algorithms and the procedure for compiling optimal ensembles of automatic grouping algorithms was built into the production process of testing the electronic component base of spacecraft at ITC-NPO PM JSC (Zheleznogorsk). It made it possible to ensure high separation accuracy into homogeneous batches of industrial products, reduce the calculation time and requirements for computing resources, as well as provide the ability to make decisions about the selection of product samples from each homogeneous batch for destructive analysis in an interactive mode.

Chapter 1. OVERVIEW OF MODERN CLUSTERING METHODS WITH INCREASED REQUIREMENTS FOR THE ACCURACY AND STABILITY OF THE RESULT

The chapter is devoted to the analysis of the current state and development of methods and problems of automatic grouping in conjunction with the problems of location theory. The problems arising when solving problems of automatic grouping of objects with the increased requirements for the accuracy and stability of the result are indicated.

1.1. General statement of clustering problems and fields of application

The modern rapid development of technologies for the automatic data collection, information transmission and storage, data mining, as well as technological growth in many industries have led to the emergence of great arrays of multidimensional data.

A lot of the data arrays have already been stored in digital form or they are being digitized intensively. At the same time, the volume and quality of modern tools and solutions, including systems for data collecting, storing and processing, is increasing. As a result, the need for their qualitative analysis and reliable conclusions for making effective management decisions is also increasing. So, it requires new achievements in the methods of the information perception, automatic processing and generalization [1, 2].

Currently, there exist some statistical methods for data analysis such as factor analysis, multivariate scaling, cluster analysis, regression, regression analysis, analysis of variance, discriminant analysis, correlation analysis [2-4].

As a rule [5], researchers distinguish the following two large classes of problems in data analysis:

1. Classification (supervised learning). They have a training sample, where data must be assigned to one or another predetermined class.

2. Clustering (unsupervised learning). A particular group (cluster) the data belong to is not known in advance, and very often the number of groups is not known as well.

In both cases, objects are divided into homogeneous groups, only the division into clusters is much more complicated. As a rule, clustering is understood as automatic grouping.

The purpose of automatic data grouping is to select such homogeneous subsets (natural groupings of objects) in the original multidimensional data so that objects within groups are to be "similar" to each other, and objects from different groups are not to be similar according to their parameters or characteristics [6].

Cluster analysis is one of the most promising areas in intensive data analytics [7]. For the first time, a term "cluster analysis" (English cluster means a bunch, bundle), according to most scientists, was proposed by the mathematician R. Trion [8]. The scope of cluster analysis is very wide. It is used in some disciplines such as archeology, medicine, psychology, chemistry, biology, government, philology, anthropology, geology, and others.

The problem of automatic grouping is formulated as follows. There are N objects; find k groups there (i.e., divide N objects into k disjoint subsets) in such a way that, based on some measure of similarity, objects belonging to the same group, were similar (had similar parameter values), and objects belonging to different groups would differ in the parameter values. This is a hard clustering problem.

The peculiarity of fuzzy clustering problem is that the partition of N objects to each of the k groups will be performed with a certain conditional probability.

This problem formulation for automatic grouping in this form leaves at least two problems unresolved, i.e., how to determine the total number of groups of objects *k,* and what measure of similarity/difference of objects to apply.

Groups can differ in size, shape and density when using the metric definition of similarity (the distance in a certain space of numerical attributes between objects). The noise presence in data makes it much more difficult to find groups.

The solution to the problem of automatic grouping is ambiguous for the following reasons [9]:

– number of groups is generally not known in advance;

– there is no best criterion for the quality of automatic grouping (without using pre-labeled data), and therefore, the partitioning may differ from case to case;

– if a problem statement includes a measure of difference, i.e., the distance between objects, then the result depends on the chosen metric or measure of a distance.

Assume that objects or data items are represented by points in some space of characteristics. Then an ideal group can be defined as a series of points that is isolated and compact. In fact, a group is an entity, the perception of which is often subjective and its definition may also require knowledge in the relevant field. In practical problems, the number of data measurements can be very large, for example, in the applied problem of grouping electrical radio products, which is given as an example in this chapter, the dimension of data can vary from several tens to thousands of measurements [10, 11].

In almost any discipline that involves multidimensional data analysis, there are clustering problems. It is difficult even to list the numerous scientific fields in which automatic grouping methods are used, as well as the many different methods and algorithms that exist.

The examples of automatic grouping tasks are document categorization to provide quick access and search [12-15], image segmentation (in computer vision) [16-19], problems of handwritten [20] and printed [21] text recognition, grouping of potential service customers points by geographic/geometric proximity for efficient service organization [22], biological tasks [23, 24], work with groups of customers (consumers) in CRM systems [25, 26].

The problem solution of the automatic objects grouping in the most common formulations, as already mentioned, presupposes the presence of a certain measure of similarity, or vice versa, a measure of difference, which, in fact, is the distance between objects in some discrete or continuous space of characteristics. For example, the reciprocal can be the measure of similarity.

1.2. Location theory and clustering problems

The automatic grouping problem in its most frequently applied definition, as a rule, operates with the locations of some points (objects) in space and distances between them, its connection with the problems of the location theory is traced. They are most often defined by scientists as problems; their main parameters are locations of some objects in space and a distance between them [27-30]. The relationship between the location theory and the problems of clustering has developed quite a long time ago [31-34], originating in the framework of economic theory [35].

Location problems are widely used both directly [27, 36-39] e.g., (in architecture, urban development, transportation, etc.) and indirectly (e.g., in standardization) [40-42]. In the USSR, starting from the 1960s, (for example, at the S.L. Sobolev Institute of Mathematics), location problems were formulated to determine the optimal composition of technical systems, optimal range of products. They were not formally directly related to location in the geometric sense [43 -45]. Subsequently, when the connection between these two directions of research was revealed, the focus of attention of scientists gradually shifted towards the location problems.

Location problems can be classified according to the dependence of the objective function on distances between new and existing objects: continuous (if the object can be placed at any point in space), discrete (if a location is possible only at certain points), problems on the network are also distinguished [46].

Clustering models often have similarities with models of the object location theory, and sometimes even identical to them, therefore, they were often considered by researchers together. The parallel development of the location theory and cluster analysis gave the same or very similar methods. For example, one of the most widespread algorithms in location theory is the ALA (Alternating Location-Allocation) procedure for solving the p-median problem [47] and the k-means procedure [48] (a rather widespread algorithm in cluster analysis) are built according to the same scheme.

Fermat's problem is probably the simplest problem of the location theory, i.e., find a new point for the given three points when a sum of distances to the known points will be minimal [49, 50, 51]. T. Heinen, F. Simpson and others also were dealing with this problem.

Later A. Weber developed Fermat's problem. A new problem requires finding the minimum point of the sum of distances already for an arbitrary number of the known points. Point weights have been added to the problem. The study was devoted to the influence of main factors of production on the location of enterprises in order to minimize costs [52]. This problem, called the Weber (Fermat-Weber) problem, the 1-median problem [53], or the Steiner problem, served as the starting point for the development of the location theory.

The procedure for solving the Weber problem is included into some methods for solving clustering problems (it is their integral part), in particular, methods based on greedy agglomerative heuristic algorithms, combinations with which are used for research in this work. A. Weissfeld in his work [54] proved the theorem formulated by Sturm [55] and defined a sequence that would converge to the optimal solution of the Weber problem, which was essentially a version of the gradient descent algorithm [56]. This algorithm, in its more perfect variation [57], is still widely used to solve location problems. S.L. Hakimi determined the possibility of discretizing the continuous Weber problem [58, 59]. Note that in the case of using the square of the Euclidean distance as a measure of distance, the Weber problem is solved in an elementary way [28]: a solution is a point whose coordinates are the averaged values of the coordinates of known points. This circumstance explains the popularity of the k-means algorithm, which uses exactly the square of the Euclidean distance.

Clustering and location problems are traditionally formulated by Russian (formerly Soviet) scientists in interrupted space and on networks by linear and integer linear programming problems [60]. The NP-complexity of most these problems have been proved, but despite this, a large arsenal of efficient solution methods has been developed for them, most of which can be classified as hard methods [40, 41, 44, 61, 62–65].

In the former USSR V.R. Khachaturov and V.P Cherenin studied the issue of enterprises location [66-69]. At the Institute of Mathematics named after S.L. Sobolev of SB RAS works by E.Kh. Gimadi, V.L. Beresnev, A.A. Kolokolov, and later Yu.A. Kochetov, A.V. Eremeev, G.G. Zabudsky, T.V. Levanova. [30, 40-43, 61, 70-72] and others in the development of models of standardization and unification were a theoretical basis for the development of algorithmic and mathematical apparatus for solving problems of automatic grouping and the theory of objects location.

If we consider automatic grouping algorithms, local search is implemented there by sequentially improving a previously known intermediate result, and therefore the dependence of the result of the algorithms on the selected initial solution is observed. Since the search for the next solution is not necessarily carried out in the neighborhood of the previous one, such algorithms in the strict sense cannot be classified as optimization methods of local search. It is possible to organize work in the multiple start (multistart) modes of these methods with randomized procedures for choosing initial solutions [73, 74] or more complex approaches [4]. Also, there exist approaches based on ideas from living nature, i.e., genetic and other evolutionary algorithms [75, 76], neural networks [77], as well as methods for simulating annealing [78], etc.

A lot of varieties of the genetic algorithm are characterized by the fact that they often receive a solution in the form of a global optimum (although the task of checking the found optimum for globality is in turn also a difficult task), while classical methods of local search easily find the actual local optima of the problem [43].

Alp O., Erkut E. and Dresner C. proposed a genetic algorithm that applies a special recombination (crossing over) procedure, i.e., a greedy (agglomerative) heuristic procedure [79]. Under a heuristic algorithm or procedure, called in the literature "heuristics", we mean an algorithm that does not have a rigorous justification, but it gives an acceptable solution to the problem for most practical cases. The paper [79] proposes an algorithm for solving the p-median problem on a network. Instead of rearranging the sequences that represent the parent "individuals", this heuristic combines the parental sets of host indexes [80, 81], selected as the centers of the groups, i.e., a child solution contains more centers than is required by the conditions of the problem. Further, there is a sequential removal of unnecessary centers (that element of the solution, the removal of which gives the smallest increase in the objective function) until a feasible solution is reached.

Evolutionary algorithms (including genetic ones) apply principles and terminology associated with natural evolution. The advantage of these algorithms over classical ones was investigated experimentally [82]. But evolutionary and genetic algorithms only determine the general scheme for organizing the search.

The implementation of the search scheme in the genetic algorithm depends on the choice of selection procedures, mutation, and especially crossing over that was originally conceived as a simple randomized procedure [83]. Later, the crossover procedure (recombination, crossing) was implemented in the form of sometimes very complex algorithms for specific optimization problems [84]. In the course of solving some NP-hard problems [85, 86], the efficiency of approaches was proved, in which randomized recombination procedures (crossing) are replaced by recombination methods optimized for specific problems. Thus, a choice of the crossing procedure is very important, although a choice of the optimal procedure can be as difficult as finding the best solution to the optimization problem (the best value of the objective function in a limited time).

1.3. Main methods of cluster analysis

Modern methods of cluster analysis offer a wide range of tools for identifying groups of different aggregate parameters. The k-means method is the most common among these methods [2, 4, 6]. The k-means algorithm itself is a local optimization algorithm, and its result depends on the selection of initial values (averaged parameters of centers or centroids of groups, i.e., clusters). At the same time, the method for identifying groups of objects with different parameters should give reproducible results.

The k-mean problem, along with a very similar p-mean problem, is one of the classical problems in location theory [28]. The k-means task is to find such k centers of clusters $X_1 \ldots X_k$ in d-dimensional space so that the sum of the squared distances from them to the given points A_i ($A_1, \ldots, A_N$) is minimal:

$$\arg\min F(X_1,\ldots,X_k) = \sum\nolimits_{i=1}^{N} \min_{j\in\{1,k\}} \left\| X_j - A_i \right\|^2 , \qquad (1.1)$$

The most widely spread method for solving the k-means problem is the k-means algorithm of the same name, also called the Alternating Location-Allocation (ALA) procedure. It is a simple and fast algorithm that can be applied to many automatic grouping and placement problems. The algorithm includes only two steps, alternating in the course of operating, i.e., splitting into groups or clusters (an object belongs to the group whose center is the closest one to it) and recalculation of the centers of the groups. The algorithm sequentially improves the known solution, making it possible to find the local minimum (1.1).

The algorithm has limitations. At the beginning of the solution, it is necessary to specify the number of groups k the objects are divided into; the result strongly depends on the initial solution, as a rule, selected randomly.

Algorithm 1.1 K-means algorithm

```
Given: data vectors $A_1...A_N$, $k$ initial cluster centers $X_1...X_k$
fulfill out
1: Build a cluster $C_i$ of data vectors for each center $X_i$ so that for each data vector its center is the nearest.
2: Calculate a new $X_i$ center value for each cluster.
while steps 1-2 lead to any changes.
```

The idea of a k-means algorithm was proposed by Steinhaus [87] in 1956. The algorithm itself was developed by Lloyd [48, 88]. Since then, the k-means algorithm (Lloyd's algorithm), its improvement, modification and combination with other algorithms have become a theme for investigations of some scientists. In the case of the classical k-means problem, cluster centers are usually called centroids.

Alsabti et al. [89] proposed an efficient clustering method applying a pattern in a k-dimensional tree. Nigam et al. [90] presented an algorithm applying tagged and untagged documents based on a classifier. Kanungo et al. [91] described the use of k-means as a filtering algorithm. Chung [92] presented a generalized k-means algorithm that gives correct clustering results without a known number of clusters. Xiaoli et al. [93] proposed an algorithm based on k-means and working not with the entire data space, but only with representative points selected using sampling. Xiong et al. [94] investigated the effect of data distribution on the k-means algorithm. They studied measures of k-means and clustering in terms of data distribution. In fact, their focus has been on characterizing the relationship between data distribution and k-means clustering in addition to the measure of entropy and measure of accuracy.

Zhang et al. [95] proposed a simple and effective methodology for classifying NBA (National Basketball Association) defenders based on the k-means algorithm and Euclidean distances as a measure of difference. Wang et al. [96] presented an improved k-means algorithm that filters noise in clustering and overcomes the disadvantages of the original method. The original algorithm has built-in steps for analyzing and processing noisy data based on density determination. Singh et al. [97] describe a modified k-means algorithm based on the sensitivity of the initial cluster centers. This algorithm divides the space into segments and calculates the frequency of the points in each segment. A segment with the highest point frequency contains the cluster center with the highest probability. The paper by Shi Na et al. [98] describes a k-means algorithm improved by steps that preserve the distance information obtained in previous iterations and save computation time. A similar approach was used by Rani [99].

An important part of the k-means algorithm is the selection of starting centers for the algorithm to work, which is often the topic of additional studies. Busare and Bansod [100] describe a k-means algorithm combined with an improved pillar algorithm. The pillar algorithm is efficient for selecting starting centers, but has outlier problems leading to reduced performance. It is possible to solve this problem improving the algorithm. The problem of selecting initial centers was also dealt with by Kaur et al. in [101]. In [102] Shunye Wang et al. applied a difference matrix constructed using the Huffman tree to select the initial centers. Mahmoud et al. [103], in the case of weighted multivariate data, applied a heuristic method to select the starting centers, which includes calculating the average and sorting by merge. Abdul Nazir and Sebastian in their work [104] described an improved k-means algorithm, which includes special methods for determining the initial centers and binding points to clusters.

When we are speaking about the k-means problem and its solution, it is necessary to mention also the j-means algorithm developed by Hansen and Mladenovic [105], and considered one of the most efficient and accurate algorithms for this problem, as well as for the p-median problem. The algorithm replaces the centers with one (the best in terms of the objective function) from the data vectors and then continues the search using standard k-means.

An algorithm for the k-medoid problem, Partitioning Around Medoids (PAM), was proposed by Leonard Kaufman and Peter J. Rousseeuw [106]. It is similar to the k-means algorithm. Both algorithms operate by trying to minimize error, but PAM works with medoids, which are objects that are part of the original set and represent the group they are included in, and k-means works with centroids, artificially created objects that represent cluster. The PAM algorithm divides a set of N objects into k clusters (N and k are the algorithm inputs). The algorithm works with a distance matrix, its goal is to minimize the distance between representatives of each cluster and its members.

The PAM procedure consists of two phases, BUILD and SWAP:

1. BUILD. Primary grouping is performed, during which *k* objects are consequently selected as medoids.

2. SWAP is an iterative process that attempts to improve multiple medoids. The algorithm searches for a pair of objects (medoid, non-medoid) that minimize the target function during replacement, and then update the set of medoids.

At each iteration of the algorithm, a pair is selected (medoid, non-medoid) such that replacing the medoid with a non-medoid gives the best possible clustering. Clustering is estimated using the objective function calculated as the sum of the distances from each object to the nearest medoid. A procedure for changing a set of medoids is repeated as long as it is possible to improve the value of the objective function.

Algorithm 1.2 PAM procedure

```
Build phase:
    1. Select $k$ objects as medoid.
    2. Build a distance matrix, if not specified.
    3. Assign each object to the nearest medoid.
Swap phase:
    4. Find objects that reduce the total distance for each cluster and if there are such objects, select those objects that reduce greatly, as a medoid.
    5. Return to step 3 if at least one medoid has changed, otherwise complete the algorithm.
```

Popular methods of automatic grouping include the Expectation Maximization algorithm (EM-algorithm means maximization of mathematical expectation) [107]. The main idea of the algorithm is to introduce artificially an auxiliary vector of hidden variables, which reduces a complex optimization problem to two steps:

1. E-step is a sequence of iterations for recalculating hidden variables according to the current approximation of the parameter vector;

2. M-step is likelihood maximization (for finding the next vector approximation).

The clustering problem solved by the EM-algorithm is reduced to the problem of separating a mixture of probability distributions. General description of the EM-algorithm (for separating a mixture of distributions) [107-109]:

Algorithm 1.3 EM -algorithm

Given: Sample (array) of N vectors of d-dimensional data $X_i = \left(x_{i,1}, \dots, x_{i,d}\right)^T, i = \overline{1,N},$ the estimated number of distributions in a mixture k.

Step 1 (initialization). Select some initial values of distribution parameters. As a rule, as vectors of mathematical expectations μ for the problem of separating a mixture of Gaussian distributions, the values of randomly selected data vectors are chosen, and the values of variances (or covariance matrices) are set the same for all distributions and are calculated for the entire sample, or unit matrices are taken as covariance matrices (similarly, for exponential or Laplace distributions, a parameter α is calculated over the entire sample $X_1,\dots,X_N$).

Set values of the prior probabilities of each distribution to be equal for all distributions $w_j = 1/k, j = \overline{1,k}.$

Step 2 (E-step - classification / clustering).

With fuzzy clustering, for each distribution j and for each data vector i, the posterior probability is calculated that the i-th data vector belongs to the j-th distribution: $g_{i,j} = \frac{f(x_i|j)w_j}{\sum_{l=1}^{k}(f(x_i|l)w_l)} \forall i = \overline{1,N}, j = \overline{1,k}.$ Here $f(x_i|j)$ is the density of the j-th distribution at the point x_i.

Step 3 (M-step is modification of distribution parameters).

3.1. Recalculate the values of prior probabilities:

$$w_j = \frac{\sum_{i=1}^{N} g_{i,j}}{N} \forall j = \overline{1,k}.$$

3.2. Recalculate parameters' evaluations of each of the distributions taking into account the posterior probability that a particular i-th data vector is included in the j-th cluster with the probability $g_{i,j}$. For example, the vector of mean values $\mu_j = (\mu_{j,1}, \dots, \mu_{j,d})$ for each cluster is calculated by the formula:

$$\mu_{j,l} = \frac{1}{\sum_{q=1}^{N} g_{q,j}} \sum_{i=1}^{N} x_{i,l} g_{i,j} = \frac{1}{Nw_j} \sum_{i=1}^{N} x_{i,l} g_{i,j} \quad \forall j = \overline{1,d}, l = \overline{1,k}.$$

Similarly, evaluations of standard deviations are calculated as follows:

$$\sigma_{j,l}{}^2 = \frac{1}{\sum_{q=1}^{N} g_{q,j}} \sum_{i=1}^{N} (x_{i,l} - \mu_{j,l})^2 g_{i,j} = \frac{1}{Nw_j} \sum_{i=1}^{N} (x_{i,l} - \mu_{j,l})^2 g_{i,j} \quad \forall j$$
$$= \overline{1,d}, l = \overline{1,k}.$$

Here $\sigma_{j,l}$ is a standard deviation in the *l*-th dimension in the *j*-th distribution (cluster).

Applying a multivariate Gaussian distribution with a full covariance matrix, we have:

$$\Sigma(j) = \begin{pmatrix} \sigma(j)_1^2 = \sigma_{1,1} & \sigma(j)_{1,2} & \dots & \sigma(j)_{1,d} \\ \sigma(j)_{2,1} & \sigma(j)_2^2 = \sigma(j)_{2,2} & \dots & \sigma(j)_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(j)_{d,1} & \sigma(j)_{d,2} & \dots & \sigma(j)_d^2 = \sigma(j)_{d,d} \end{pmatrix}$$

Its elements are also calculated taking into account posterior probabilities:

$$\sigma(j)_{p,q} = \sigma(j)_{q,p} = \tfrac{1}{Nw_j} \textstyle\sum_{i=1}^{N} \left(x_{i,p} - \mu_{j,p}\right) \left(x_{i,q} - \mu_{j,q}\right) g_{i,j}.$$

4. Calculate the value of the objective function, the logarithmic likelihood function:

$$Q(w_1, \dots, w_1, parameters\ of\ all\ distributions)$$
$$= \sum_{i=1}^{N} ln(\sum_{j=1}^{k} w_j f(x_i|j))$$

5. Check stop conditions, then go to Step 2.

The following conditions are used as stop conditions:

A) reaching the limit on the number of *ITER* iterations;

B) reaching the limit time of the algorithm t_{max};

C) No change in the value of the objective logarithmic likelihood function.

Note that the result of the EM algorithm is the probability matrix $g_{i,j}$, each element of which means the probability that the *i*-th object belongs to the *j*-th cluster (i.e., generated by the *j*-th distribution).

Classification Expectation Maximization (CEM) [108, 110] is a modification of the EM algorithm works on the principle of a clear classifier of sample data. In this case, each object belongs to a single cluster. The CEM algorithm almost coincides with another modification, SEM (Stochastic EM) [111, 112, 113], only the first one at each step introduces a deterministic rule that the data refer to only one cluster for which the maximum posterior probability was calculated. Thus, the CEM algorithm, in contrast to EM, solves the problem of clear clustering.

As it was noted in the previous section, the connection between automatic grouping problems and location theory problems is obvious. Moreover, the most popular algorithms for solving such problems as k-means, k-medoid and EM-algorithm are similar in structure. Each of them contains two alternating steps. In all these problems, objective functions have the property of being multi-extremal. The generality of the properties of such models and the corresponding algorithms gives a reasonable hope that similar methods of increasing the accuracy and stability of the obtained results of solving problems will be effective.

For further presentation in this study, each of the procedures, i.e., k-means, k-medoid, and CEM will be denoted as a two-step local search algorithm. Moreover, the k-means and k-medoid procedures themselves are, in fact, search algorithms with alternating neighborhoods. Further, when solving k-means problems, as a two-step local search algorithm, it will be implemented by Algorithm 1.1, respectively it will be implemented, for k-medoid it will be implemented by Algorithm 1.2 and maximizing the likelihood function it will be implemented by the CEM-algorithm.

The directed search for possible solutions in a relatively small subset of space is performed in the considered classical algorithms of automatic grouping (for example, k-means, k-medoid). It does not guarantee finding a strictly optimal solution when we apply various constraints (number, shape of the resulting groups, etc.). Despite the fact that there are a lot of methods for solving problems of automatic grouping based on classical models [114] that claim to find a globally optimal solution (practically inapplicable to very large problems and do not guarantee an exact solution), nevertheless, the main direction of modern research is in the development of heuristic methods and algorithms that find suboptimal solutions, but at the same time close to the true optimum of the problem [115, 116]. Thus, the known algorithms do not show the best results, especially in a limited fixed time when solving problems of automatic grouping with increased requirements for the accuracy and stability of the result.

1.4. Example of a relevant clustering problem with increased requirements for the accuracy and stability of the result

The relevance of solving problems of automatic grouping with the increased requirements for the accuracy and stability of the result is due to the range of their application, both in the tasks of cluster analysis and directly in practical problems in production where high accuracy of separation into homogeneous batches of industrial products is required. Consider one of the practical examples.

Orlov V.I. and Sergeeva N.A. in [117] said: “A modern spacecraft is a complex electronic system, which, being in space for 10-15 years, must diagnose itself, check, make a decision within the assigned tasks and perform various functions assigned to it. Space is an aggressive medium with various destructive characteristics. Deep vacuum, large temperature difference, radiation, flows of charged particles, etc. are among them. The onboard equipment in space cannot be repaired. That is why it is called non-repairable, and, accordingly, the reliability of such equipment should be maximized. The required reliability level is ensured by various factors. The most important of them is the application of highly reliable electronic components. The spacecraft (SC) contains from 100 to 200 thousand electronic components (EC). They include microcircuits, transistors, diodes, capacitors, relays, quartz resonators, resistors, etc. Each year the overall dimensions of electronic components are getting smaller, and a degree of integration of microcircuits is getting higher. The dimensions of leadless resistors or capacitors reach 1–2 mm, and their weight is less than a gram. Integrated microcircuits of the processor type have the capabilities of a personal computer packed in a 5×5 cm case. The equipment of the spacecraft onboard equipment with a highly reliable EC is one of the main problems of the modern space industry. First of all, it is necessary to prevent low-grade counterfeit products that do not meet the requirements for the reliability from entering the equipment. For this problem solving, it is necessary to ensure the purchase of electronic components from trusted suppliers, as well as conducting incoming control (IC), additional screening tests (AST) and destructive physical analysis (DPA) of electronic components. Individual component analysis is of the particular importance".

Unlike the United States of America and Western European countries, there are no specialized EC manufactures for the space industry in our country. Therefore, electronic, electrical, electromechanical parts (EEE parts) must be tested for use in spacecraft equipment. For this purpose, for many years in Russia there has been a principle of completing spacecraft equipment through specialized test technical centers [118, 119] with various EEE tests (IC, AST, XRF and DNE - diagnostic nondestructive evaluation).

This approach immediately yielded real results. So, if the majority of the spacecrafts of JSC "ISS" (Zheleznogorsk) operated before 2000 had comments on the quality of functioning, starting from the first days or months of operation, then significant remarks to ERI for almost 20 years of operation on the spacecraft "Sesat" operated since April 2000. According to the opinion of the majority of experts, it happened due to the fact that for the first time in practice all 100 percent of the EEE parts of the Sesat spacecraft passed IC, DPA, DNE and XRF [119-121].

Various scientific papers [122-128] set forth, for example, the tasks of ensuring the radiation resistance of EEE parts. But they are based on the assumption that the radiation resistance of any electrical radio product from a production batch is known and, most importantly, the same. In practice, the characteristics of EEE parts (including radiation resistance) are different and they depend on different reasons [129] [119-121].

It is necessary to be sure that we are dealing with a batch of electrical radio products made from a single (homogeneous) batch of raw materials in order to extend the test results to the entire production batch of products. Therefore, the identification of homogeneous production batches from prefabricated EEE parts batches should become one of the most important stages in testing. In practice, a lot of tests are carried out, from tens to several thousand for each product; the results are tabulated and serve as data for the analysis. The procedure for separating the parameters should be regulated. The repeated calculation of the data should give the same or very close results. According to the information given above, a task is formed as follows: separation into homogeneous production batches based on test data (clustering). All data collected during deviation tests are applied. It means that clustering is performed in the space from tens to hundreds of thousands of vectors [130].

Modern methods of cluster analysis offer a wide range of tools for identifying groups of different aggregate parameters. At the same time, a method of identifying groups (clusters) of electrical radio products with different parameters should give reproducible results. The algorithms proposed in Chapters 2 and 3 help to increase the accuracy of automatic grouping methods to identify groups of electrical radio products of different parameters.

1.5. Variable neighborhoods search method

Local search methods have been further developed in metaheuristics (optimization methods that reuse simple rules or heuristics to achieve an optimal or suboptimal solution, characterized by greater stability) [131]. Consider the one, called Variable Neighborhoods Search (VNS), a popular method for solving discrete optimization problems by N. Mladenovich and P. Hansen. It helps to find good suboptimal solutions to fairly large problems of automatic grouping [132, 133, 134]. The main idea is to systematically change of the neighborhood type while the local search.

There exist a lot of options for implementing the Variable Neighborhood Search method for large-scale problems. The flexibility and high efficiency explain its competitiveness in solving NP-complexity problems. It is reflected in papers presented by Yu.A. Kochetov, F.G. Lopez, J. Brimbern, T.V. Levanova, E.V. Alekseeva and others, in particular, for solving problems of automatic grouping and location [30, 61, 70, 71, 135, 136], Weber's multiple problem [137], the p-median problem [72, 138] and many others.

We denote the finite set of types of neighborhoods previously selected for local search as N_k, k=1,..k_{max}. The proposed method with variable neighborhoods is based on the fact that a local minimum in one neighborhood is not obligatory a local minimum in another neighborhood, while the global minimum is local in any neighborhood [139]. Moreover, on average, local minima are closer to the global than a randomly selected point, and they are located close to each other. This helps to narrow the search area for the global optimum applying the information about the already discovered local optima. This hypothesis underlies various crossover operators for genetic algorithms [140] and other approaches.

The implementation of the local Variable Neighborhood Search is possible in one of three ways, i.e., deterministic, probabilistic, or mixed, combining the two previous ones [139].

The Variable Neighborhoods Descent (VND) assumes a fixed order of changing neighborhoods and the search for a local minimum with respect to each of them. The Reduced Local Descent with Variable Neighborhoods (RVNS) differs from the previous VND method by a random selection of points from the neighborhood *$O_k(x)$*. A stage of finding the best point in the neighborhood is omitted. The RVNS algorithms are most productive when solving problems of large dimensions. In this case the application of the deterministic variant requires too much operating time for each iteration.

The basic scheme of Variable Neighborhood Search (VNS) is a combination of the two previous variants (VND and RVNS) [133].

Algorithm 1.4 VNS **(**Variable Neighborhood Search)

```
Step 1. Select the neighborhood $O_k$, $k$=1,…,$k_{max}$, and a starting point $x$.
Step 2. Repeat until the stop criterion is fulfilled.
2.1. $k$←1;
2.2. repeat until $k \le k_{max}$;
2.2.1. select a point $x' \in \mathrm{O}_k(x)$ randomly;
2.2.2. apply the local descent from the starting point $x'$ without changing coordinates when $x$ and $x'$ coincide. Denote the obtained local optimum as $x''$;
2.2.3. if $F(x'') < F(x)$, then we set $x \leftarrow x''$, $k \leftarrow 1$, otherwise $k \leftarrow k + 1$.
```

The essence of the Variable Neighborhoods Search Algorithm [132] is that for some intermediate solution, a set of neighborhoods of this solution is determined. The next type of neighborhood is selected from this set. The corresponding local search algorithm is applied for its search. If this algorithm finds an improved solution, the intermediate solution is replaced by a new solution. The search continues in the same neighborhood. If the next local search algorithm could not improve the solution, a new search neighborhood is selected from a set of neighborhoods of the intermediate solution.

The stop criterion can be the maximum computation time or the maximum number of iterations without changing the best-found solution.

When solving large-scale problems, the complexity of performing one iteration becomes very large, and new approaches are required to develop efficient methods of local search.

1.6. Development of the greedy heuristics method for clustering problems

The main distinguishing feature of greedy agglomerative heuristic methods is that, being methods of local search (consistently improving a known result) in a certain neighborhood of a known solution, they choose as the next solution the option that gives the greatest decrease in the value objective function (the largest increase in value is in the case of maximization).

The method of greedy heuristics was proposed by L.A. Kazakovtsev and Antamoshkin A.N. for automatic grouping problems on models of the location theory [141, 142]. Algorithms of this method obtain results for practical problems that are difficult to improve significantly by other methods in a comparable time. Despite the fact that the algorithms of the greedy heuristic method are mostly randomized; results they receive are quite stable, i.e., they give very similar results when one restarts a system. The main peculiarity of greedy agglomerative heuristic methods is that, being methods of local search (consistently improving a known result) in some neighborhood of a known solution, they choose as the next solution the option that gives the largest decrease in the value of the objective function (the largest increase in the value is in the case of maximization).

The greedy heuristics method applies evolutionary algorithms as one of the possible ways to organize a global search, including the approaches of the Krasnoyarsk school of evolutionary algorithms headed by Semenkin E.S. [143-146].

The *k*-means and *p*-median problems, except for special cases, are NP complex. They require a global search. The problem is that the result depends on the choice of the initial cluster centers from the point of view of ensuring the reproducibility of the calculation results.

A wide range of global search strategies can be applied.

The greedy heuristics method can be represented as three nested loops for automatic grouping problems when dividing a set of objects into *k* groups (clusters):

1) A cycle that iterates some global search strategy generates intermediate solutions represented by sets with cardinality greater than *k*.

2) A greedy heuristic procedure is fulfilled. The local search algorithms can be triggered; they lead intermediate solutions to feasible ones and at the same time they improve intermediate solutions.

3) A local search cycle evaluates the consequences of excluding elements from an intermediate solution.

Figure 1.1 presents a block diagram of the greedy heuristic method in its most general form [142].

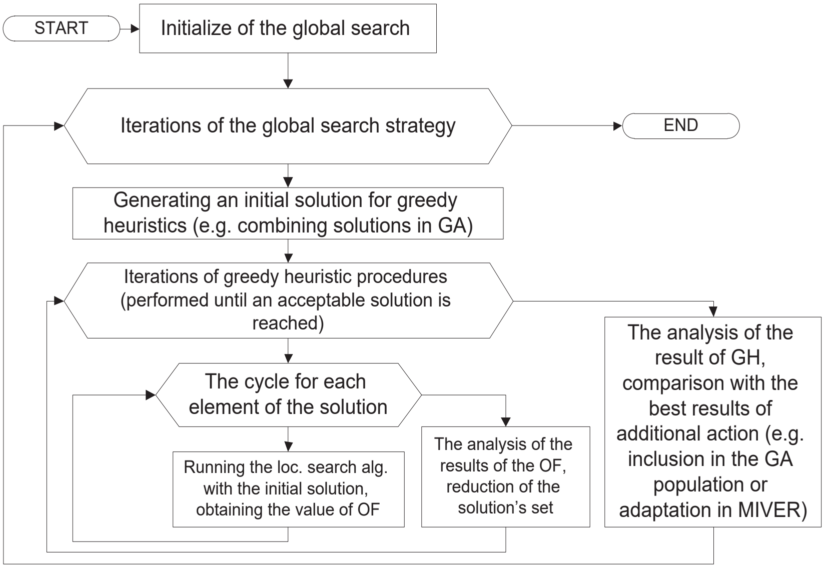

START
Initialize of the global search
Iterations of the global search strategy
END
Generating an initial solution for greedy heuristics (e.g. combining solutions in GA)
Iterations of greedy heuristic procedures (performed until an acceptable solution is reached)
The cycle for each element of the solution
Running the loc. search alg. with the initial solution, obtaining the value of OF
The analysis of the results of the OF, reduction of the solution's set
The analysis of the result of GH, comparison with the best results of additional action (e.g. inclusion in the GA population or adaptation in MIVER)

Fig. 1.1. General scheme of the method of greedy heuristic algorithm

The diagram in Figure 1.2 reflects the mutual compatibility of various problem statements (discrete/continuous location problem or grouping, pseudo-Boolean monotone optimization problem), various global search strategies (MIVER/GA/multistart/deterministic methods), various auxiliary local search methods that are effective when the solution of a particular problem, as well as the used distance measures when they are applied as part of the greedy heuristic method [142].

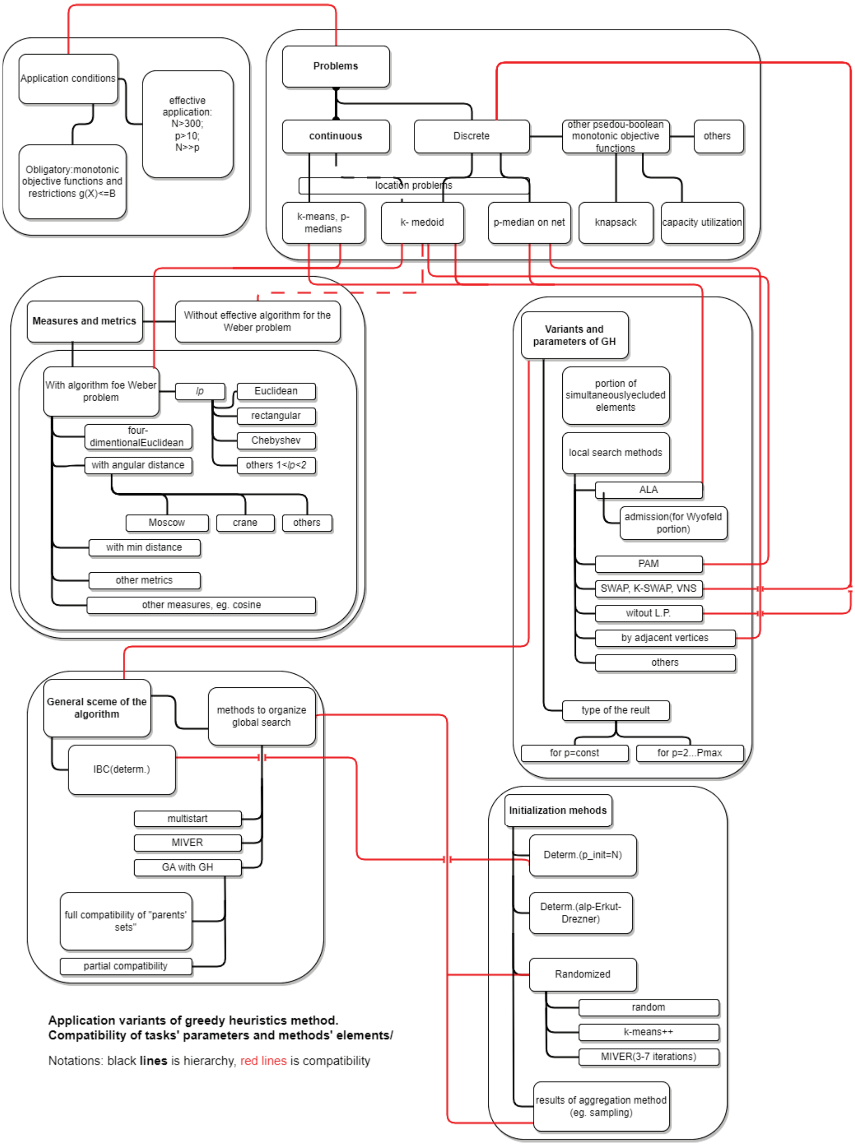
The algorithm of the basic greedy agglomerative heuristic procedure is presented below:

Algorithm 1.5 Basic Greedy Agglomerative Heuristic Procedure (Greedy)

```
Given: the initial number of clusters $K$, the required number of clusters $k < K$, the initial solution $S=\{X_1,...,X_k\}$.
            1: Send the solution $S$ to a two-step local search algorithm,
               get a new, improved solution $S$.
  while $K \neq k$
      for each $i' \in \{\overline{1,K}\}$
          2.1: $S' = S \setminus \{X_{i'}\}$.
          2.2: Send the solution $S'$ to a two-step local search algorithm,
               perform 1 to 3 iterations of the algorithm, save the obtained
               value (a value of the objective function) in $F'_{i'}$.
  end of the cycle
      3: $i'' = \arg\max_{i'=\overline{1,k}} F_{i'}$.
4: Get a solution $S'' = S \setminus \{X_{i''}\}$, improve it with a two-step local search algorithm.
  end of the cycle
```

The greedy agglomerative heuristic for *k*-means and similar problems includes two steps. Let there be two known (parent) solutions to the problem (the first of them, for example, is the best known), represented by the sets of cluster centers *S*.

A new approach to the development of automatic grouping algorithms is proposed, based on parametric optimization models, with the combined use of search algorithms with alternating randomized neighborhoods and greedy agglomerative heuristic procedures outlined in Chapters 2 and 3. It is applied to improve the accuracy of automatic grouping methods applying a greedy heuristic method for automatic grouping problems with increased requirements for the accuracy and stability of the result.


Application conditions
effective application: N>300; p>10; N>>p
Obligatory:monotonic objective functions and restrictions g(X)<=B
Problems
continuous
Discrete
other psedou-boolean monotonic objective functions
others
location problems
k-means, p-medians
k- medoid
p-median on net
knapsack
capacity utilization
Measures and metrics
Without effective algorithm for the Weber problem
With algorithm foe Weber problem
lp
Euclidean
rectangular
Chebyshev
others 1<lp<2
four-dimentionalEuclidean
with angular distance
Moscow
crane
others
with min distance
other metrics
other measures, eg. cosine
Variants and parameters of GH
portion of simultaneouslyecluded elements
local search methods
ALA
admission(for Wyofeld portion)
PAM
SWAP, K-SWAP, VNS
witout L.P.
by adjacent vertices
others
type of the reult
for p=const
for p=2...Pmax
General sceme of the algorithm
methods to organize global search
IBC(determ.)
multistart
MIVER
GA with GH
full compatibility of "parents' sets"
partial compatibility
Initialization mehods
Determ.(p_init=N)
Determ.(alp-Erkut-Drezner)
Randomized
random
k-means++
MIVER(3-7 iterations)
results of aggregation method (eg. sampling)

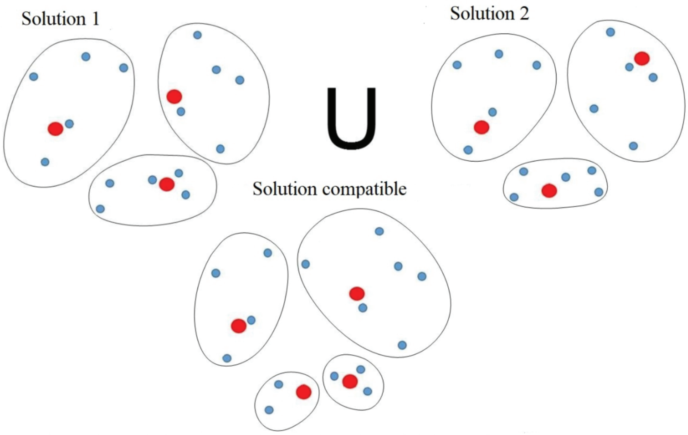
Application variants of greedy heuristics method.
Compatibility of tasks' parameters and methods' elements/

Notations: black **lines** is hierarchy, red lines is compatibility

Fig. 1.2. Scheme of the component's compatibility of the greedy heuristic method [142]

First, sets parent decisions are combined. We get an intermediate unacceptable (with an excessive number of clusters) solution (Figure 1.3).

Then there is a sequential decrease in the number of centers. Each time, centers which removal gives the least significant deterioration in the value of the objective function, are cut off.



$$Solution' = Solution1 \cup subset\ (Solution2)$$
$$result = greedy(Solution')$$

Fig. 1.3. Principle of combining parents’ solution sets

* * *

The analysis of the scientific literature showed that the methods of automatic grouping of objects developed in many ways indipendently despite the similarity with the methods of solving problems of the location theory. Popular algorithms for automatic grouping of objects are based mainly on local search heuristic methods and global search strategies.

The selection of greedy agglomerative heuristic algorithms as a tool for local search is due to the fact that these methods help obtain high-precision results. Although they require significant computational costs. These algorithms are deterministic procedures by themselves. That is why it is possible to obtain more stable results applying heuristic algorithms as a part of various global search strategies, including randomized ones.

The authors in this study set the task to develop improved algorithms for the greedy heuristic method for automatic grouping problems with the combined use of search algorithms with alternating randomized neighborhoods, which are subject to high requirements for the accuracy and stability of the result. It is necessary to take into account a certain deficit of automatic grouping methods that are compromise in the quality of the result and the computation time (by quality the authors mean accuracy, i.e., the proximity of the value of the objective function to the global optimum and stability and the proximity of the obtained values to each other with multiple runs of the algorithm).

The analysis of scientific works in this field gives reasonable hope that the proposed problem of developing new algorithms will find experimental confirmation. Moreover, the potential inherent in the method of greedy heuristics will be fully realized. Chapters 2 and 3 are devoted to the development.

Chapter 2. ALGORITHMS APPLYING GREEDY AGGLOMERATIVE HEURISTIC PROCEDURES WITH ALTERNATING NEIGHBORHOOD FOR THE K-MEANES PROBLEM

The chapter is devoted to the development of combined algorithms for the method of greedy heuristics for automatic grouping problems with increased requirements for the accuracy and stability of the result, with the joint application of search algorithms with alternating randomized neighborhoods, as well as parallel greedy heuristic algorithms for automatic grouping for massively parallel systems in relation to the problem k- mean.

2.1. Greedy agglomerative heuristics

Algorithms of the greedy heuristic method, including modifications of the greedy agglomerative heuristic procedure as a part of various global search schemes, in combination with the conditions of their applicability to certain problems, are an effective method for solving optimization problems of automatic grouping, placement, as well as problems of pseudo-Boolean optimization with a large volume of input data depending on the conditions and parameters of the problems to be solved [142]. The method of greedy heuristics itself consists in compiling an efficient combination of components when building an automated system.

Figure 2.1 presents a block diagram of the components of the greedy heuristic method and the way it can be applied [142]. The vertical order of the components reflects the nesting of the algorithms. The diagram also demonstrates the applicability of local and global search tactics to different classes of problems.

Moreover, to running a two-step local search algorithm, the most computationally demanding part of the greedy agglomerative heuristic, with a large number of clusters, is step 3, at which Algorithm 1.5 calculates the total distance after removing one cluster, i.e., $F'_{i'}$ = $F(S')$, where $S' = S \setminus \{X_{i'}\}$.

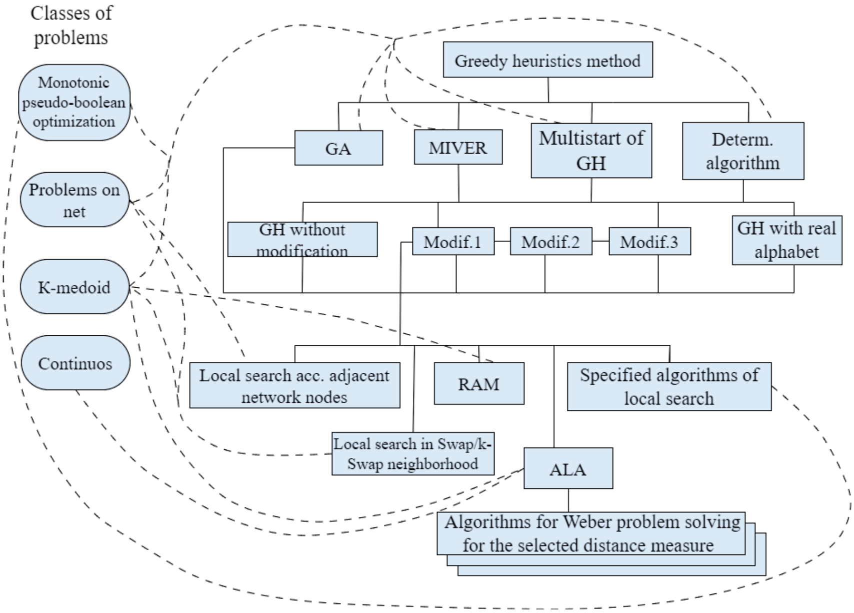

Classes of problems
Monotonic pseudo-boolean optimization
Problems on net
K-medoid
Continuos
Greedy heuristics method
GA
MIVER
Multistart of GH
Determ. algorithm
GH without modification
Modif.1
Modif.2
Modif.3
GH with real alphabet
Local search acc. adjacent network nodes
RAM
Specified algorithms of local search
Local search in Swap/k-Swap neighborhood
ALA
Algorithms for Weber problem solving for the selected distance measure

Fig. 2.1. Components of the greedy heuristic method, their mutual compatibility (solid lines) and applicability to problem classes (italic lines)

The performance of Algorithm 1.5 with large volume of data necessary for calculations becomes a problem, especially when it is possible to find the correct parameter k (a number of clusters) in practical problems only by performing several runs with a different number of clusters. Algorithm 1.5 at step 2 starts to work more and more slowly (the algorithm requires more and more iterations, and each iteration requires increasing computational resources) with an increase in the number of clusters. So, the authors made changes and implemented the removal of clusters not one at a time, but several at a time during one iteration.

Algorithm 2.1 Basic greedy agglomerative heuristic for problems with a large number of clusters

```
Given: the initial number of clusters $K$, the required number of clusters $k$ $<K$, $k>$ 50, the initial solution $S$, $|S|=K$.
    1: Improve solution $S$ with a two-step local search algorithm (if it is
       possible).
    while $K \neq k$
        for each $i' \in \{\overline{1,K}\}$
        2: $S' = S \setminus \{X_{i'}\}$. Calculate $F'_{i'}$=$F(S')$, where $F(.)$. Is the value of the objective function (for example, (1.1) for the k-means problem).
end of the cycle
        3. Set $S_{elim}$ from $n_{elim}$ centroids, $S_{elim} \subset S$, $|S_{elim}|=n_{elim}$ with minimum values $F'_{i'}$. Here, $n_{elim}=\max\{1, 0.2\cdot(|S|-k)\}$.
        4: Get a new solution $S=S \setminus S_{elim}$, $K=K-1$, and improve it applying a two-step local search algorithm.
end of the cycle
```

The algorithm of the basic greedy agglomerative heuristic procedure formed the basis of three algorithms. Similar greedy agglomerative heuristic procedures have been used as crossing over operators in evolutionary (genetic) algorithms of the greedy heuristic method [142, 147-149]. In the present study, these procedures form the neighborhoods used to modify the known solution when searching in the search algorithm with alternating neighborhoods. The proposed new heuristic procedures modify the known solution using the second known solution (Algorithms 1.5 and 2.1).

Algorithm 2.2 Greedy Procedure 1

```
Given: sets of cluster centers $S'=\{X'_1,\dots,X'_k\}$ and $S''=\{X''_1,\dots,X''_k\}$
    For each $i' \in \{\overline{1,K}\}$
        1: Combine $S'$ with a set element $S''$: $S = S' \cup \{X''_{i'}\}$
        2: Run the basic greedy agglomerative heuristic (Algorithm 1.5 or 2.1) with $S$ as the initial solution. Save the obtained result (the obtained set, as well as the value of the objective function).
        3: Return as the best solution obtained at step 2 a result (by the value of the objective function).
    end of the cycle
```

A variant where the sets are partially combined is possible. Meanwhile, the first set is taken completely; and a random number of elements from the second set is selected at random [147, 150].

```
Algorithm 2.3 Greedy Procedure 2
Given: see Algorithm 2.2.
  1: Choose random $r' \in [0;1)$. Assign $r=[(k/2-2)\,r'^2]+2$.
     Here [.] is an integer part of the number.
  2: for $i$ from 1 to $k-r$
     2.1: Generate a randomly selected subset $S'''$ of elements of the set
          $S''$ of cardinality $r$. Combine sets $S = S' \cup S'''$.
     2.2: Run a basic greedy agglomerative heuristic (Algorithm 1.5 or
          2.1) with these combined sets as the initial solution.
  end of the cycle
  3: Return the best result (based on the objective function value) from the solutions obtained at step 2.2.
```

A simpler version of Algorithm 2.2 with complete union of the sets is presented below.

```
Algorithm 2.4 Greedy Procedure 3
Given: see Algorithm 2.2.
  1: Combine sets $S = S' \cup S''$.
  2: Run the basic greedy agglomerative heuristic (Algorithm 1.5 or 2.1) with S as the initial solution.
```

These algorithms can be applied as a part of various global search strategies, and as the neighborhoods in which the search for a solution is performed, sets of solutions are used, derivatives (“off-springs”) with respect to the solution S’, formed by combining its elements with elements of some solution S’’and applying the basic greedy agglomerative heuristic procedure (Algorithm 1.5 or 2.1).

The next section considers the application of approaches of the greedy heuristic method for automatic grouping problems with a combined application of search algorithms with alternating randomized neighborhoods.

2.2. Principle of operation of combined search algorithms with alternating randomized neighborhoods for the k-means problem

Figure 2.2 presents the basic scheme of the local search with alternating neighborhoods (Algorithm 1.4).

Algorithms 2.2-2.4 can search in the vicinity of the known intermediate solution *S'*, where solutions belonging to the neighborhood are formed by adding elements of another solution *S''* and then removing the “extra” cluster centers by a greedy agglomerative heuristic procedure. Thus, the second solution S'' is a parameter of the neighborhood chosen at random (randomized) [153].

The automatic grouping algorithm for the *k*-mean problem with the combined application of alternating randomized neighborhood search algorithms and greedy agglomerative heuristic procedures can be described as follows:

Algorithm 2.5 k-GH-VNS (Greedy Heuristic in the Variable Neighborhood Search) for the k-means problem

```
1: Get solution S by running Algorithm 1.1 from a randomly generated
   initial solution.
2: $O=O_{start}$ is a search neighborhood number.
3: $i=0, j=0$.
while $j < j_{\max}$
    while $i < i_{\max}$
        4: if the STOP conditions are not met, then get the solution S'
           by running Algorithm 1.1 from the random initial solution.
        repeat
            5: Depending on the value of O (the possible values are 1, 2
               or 3), run the Algorithm Greedy procedure 1, 2 or 3, respectively, with the initial solutions S and S'. Thus, the
               neighborhood is determined by the method of including
               cluster centers from the second known solution and the
               neighborhood parameter, i.e., the second known solution.
               if the new solution is better than S, then write the new
                   result to S, $i=0, j=0$.
            otherwise leave the loop.
        end of the cycle
        6: $i=i+1$.
   end of the cycle
   7: $i$=0, $j$=$j$+1, $O$=$O$+1, if $O$>3, then $O$=1.
end of the cycle
```

In this algorithm i_{max} is the number of unsuccessful searches in the neighborhood, and j_{max} is the number of unsuccessful switches of the neighborhoods. The values of these two parameters are important in the calculations. We used the values: i_{max}=2k, j_{max}=2.

Computational experiments have shown that the Ostart parameter, which specifies the number of the neighborhood from which the search begins, is especially important. Computational experiments were carried out with all its values. Depending on the value of the O_{start}, parameter, the versions of the algorithms are designated k-GH-VNS1, k-GH-VNS2, k-GH-VNS3. Note that the search algorithm can start from different neighborhoods.

The way to obtain the second solution S' at step 4 is of great importance. By default, the second solution contains the number of centers equal to the number of centers in the solution *S*. The authors also applied modifications of Algorithm 2.5 where the number of centers in the solution *S'* is chosen randomly from the set $\{\overline{2,|S|}\}$, where |S| Is the number of centers in the solution S. In this case, the algorithms are called as k-GH-VNS1-RND, k-GH-VNS2-RND, k-GH-VNS3-RND.

Note that for *k*-means problems, the *j*-means procedure [154] is very effective, the scope of which is limited to rather small problems (the number of data vectors is usually less than 1000) [142]. The neighborhood of the current solution is determined by all possible replacements of the centroid with an object with subsequent corresponding changes. The movement is carried out in such surroundings until a local optimum is reached. This procedure is reduced to replacing the centers with one (the best from the point of view of the objective function) from the data vectors, followed by the continuation of the search using the standard *k*-means algorithm.

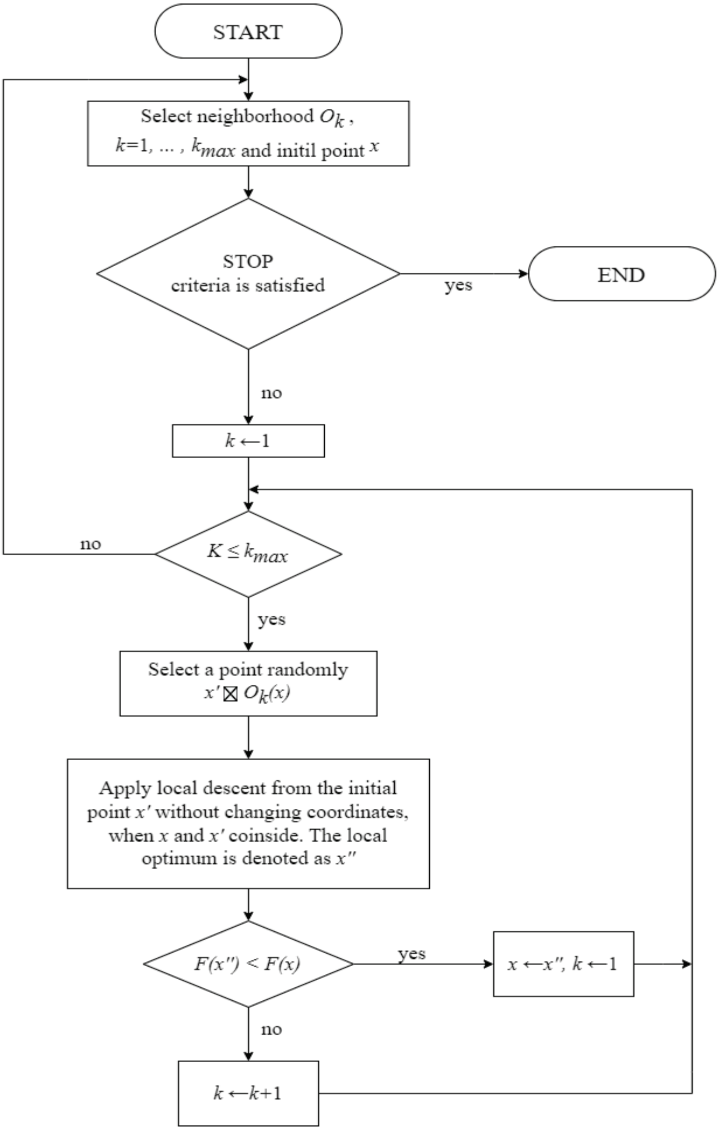

START
Select neighborhood O_k ,
k=1, ... , k_{max} and initil point x
STOP
criteria is satisfied
yes
END
no
$k \leftarrow 1$
no
$K \leq k_{max}$
yes
Select a point randomly
x' ☒ $O_k(x)$
Apply local descent from the initial point x' without changing coordinates, when x and x' coinside. The local optimum is denoted as x''
$F(x'') < F(x)$
yes
$x \leftarrow x''$, $k \leftarrow 1$
no
$k \leftarrow k+1$

Fig. 2.2. Block diagram of the VNS algorithm

Algorithm 2.6 j-means

```
Given: initial solution $S$, represented as a set of centers.
    cycle
        1: Pass the solution $S$ to Algorithm 1.1 as an initial solution.
        2: if at Step 1 the value of the objective function has not improved,
           then STOP and return the solution $S$.
        3: Generate an array $I=\{\overline{1,p}\}$, arrange the array elements in random
           order. These actions speed up the procedure.
        for all $i'\in I$
            4.1: $i=I_i', f'=+\infty$.
            4.2: $j^{'}=\arg\min_{j\in\{\overline{1,N}\}} F((S\setminus\{S_i\}\cup\{A_j\}))$, where $S_i$ is the $i$-th centroid
                 in the solution, $A_j$ is the $j$-th data vector.
            4.3: if $F((S\setminus\{S_i\}\cup\{A_j\}))<F(S)$, then $S=S\setminus\{S_i\}\cup\{A_j\}$ and complete
        end of the cycle
    end of the cycle
```

For computational experiments, we applied combined versions of the k-GH-VNS and j-means algorithms, designating them as j-means-GH-VNS (Algorithm 2.7). It already contains not three values of the neighborhood number O (as in Algorithm 2.5), but four. If the value is 4, then run Algorithm 2.6 (j-means) otherwise Algorithms 2.2-2.4.

Algorithm 2.7 j-means-GH-VNS (combined k-GH-VNS algorithm with j-means)

```
    1: Get solution $S$ by running Algorithm 1.1 from a randomly generated
       initial solution.
    2: $O=O_{start}$ (by the number of the search area)
    3: $i=0, j=0$.
    while $j<j_{\max}$
        while $i<i_{\max}$
            4: if the STOP conditions are not met, then get the solution S'
               by running Algorithm 1.1 from the random initial solution
            repeat
            5: Depending on the value of $O$ (the possible values are 1, 2,
                3 and 4), run Algorithm 2.2, 2.3, 2.4 or 2.6, respectively,
                with the initial solutions $S$ and S'. Thus, the neighborhood
                is determined by the method of including cluster
                centers from the second known solution and the neighborhood
                parameter, i.e., the second known solution.
            if a new solution is better than $S$, then
                write the new result to $S$, $i$=0, $j$=0. S, i = 0, j = 0.
            otherwise leave the loop
        end of the cycle
        6: Increment $i$.
    end of cycle
  7: $i$=0, $j$=$j$+1, $O$=$O$+1, if $O$>4, then $O$=1
end of the cycle
```

The j-means algorithm is quite "hard" in terms of computations and can take a very long time to search for a solution with a large volume of data and a large number of clusters. This property is not so important on small datasets. It takes a lot of time and resources on large datasets. The j-means-GH-VNS3 combination requires especially large computational resources, where Greedy Procedure 3 (Algorithm 2.4) with full union of sets is applied. It requires the largest computational resources among the new procedures used to form search neighborhoods. Therefore, this combination, as shown by computational experiments, does not provide significant advantages over the k-GH-VNS3 algorithm; its results are not presented.

2.3. Results of computational experiments with new algorithms for the k-means problem

The authors applied classical datasets from the UCI (Machine Learning Repository) [155] and Clustering basic benchmark [156] repositories to test our new algorithm (k-GH-VNS) in three different modifications. The datasets were selected from various spheres of life and various volumes of data for the correctness of the research:

- Ionosphere (351 data vectors, each dimension 35), i.e., classification of radar returns from the ionosphere (prediction of high-energy structures in the atmosphere from antenna data);

- Mopsi-Joensuu (6014 data vectors, each with a dimension of 2), i.e., the location of users until 2012 (Joensuu is a city in eastern Finland);
- Chess (3196 data vectors, each dimension 36), i.e., chess problems (King + Rook vs. King + Pawn);
- Europe (169309 data vectors, each with dimension 2), i.e., classification of European skills, competencies, qualifications and professions;
- BIRCH3 (100,000 data vectors, each dimension 2), i.e., clusters of random size in random locations;
- KDDCUP04BioNormed (65536 data vectors, each dimension 74), i.e., biological dataset [157].

The UCI Repository (Irvine, California, USA) is the largest repository of model and real machine learning problems based on real data on applied problems in physics, engineering, biology, medicine, sociology, etc. The data sets of this particular repository are most often used by researchers for empirical analysis of machine learning algorithms [158]. Clustering basic benchmark repository School of Machine Learning of the University of Eastern Finland (Joensuu).

For the experiments, the authors applied the Depo X8Sti computing system (6-core Xeon X5650 2.67 GHz CPU, 12 GB of RAM), the hyper-threading technology was disabled. The experiments were also carried out on a low-power system with a 2-core Atom N270 1.6 GHz CPU, 1 GB of RAM (the execution time increases 16-25 times, i.e., to achieve the same results, it takes 16-25 times longer fixed time).

Thirty attempts were made to run each of the algorithms for all datasets. Only the best results achieved in each attempt were recorded, then from these results for each algorithm the values of the objective function were calculated: minimum and maximum values (Min, Max), mean value (Mean) and standard deviation (RMSD). The j-means and k-means algorithms were run in multi-start mode.

The results of our computational experiments are presented in Tables 2.1-2.6. The best objective function values (minimum, mean, and standard deviation) are shown in bold italic type.

Table 2.1

Results of computational experiments on the ionosphere dataset (30 seconds, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
		10 clusters		
j-means	1 590,34	1 598,83	1 594,77	2,41
k-means	1 590,93	1 598,61	1 595,28	2,47
k-GH-VNS1	***1 586,38***	1 586,65	***1 586,52***	***0,12***
k-GH-VNS2	***1 586,38***	1 591,99	1 588,00	2,36
k-GH-VNS3	***1 586,38***	1 591,99	1 587,24	1,57
j-means-GH-VNS1	***1 586,38***	1 586,63	***1 586,44***	***0,10***
j-means-GH-VNS2	***1 586,39***	1 586,63	***1 586,48***	***0,12***
		20 clusters		
j-means	1 282,18	1 299,13	1 291,92	4,83
k-means	1 286,30	1 310,54	1 301,98	5,81
k-GH-VNS1	1 239,16	1 259,56	***1 246,39***	5,18
k-GH-VNS2	1 243,94	1 263,11	1 252,26	4,96
k-GH-VNS3	***1 238,53***	1 265,28	1 252,53	6,47
k-GH-VNS1-RND	***1 237,58***	1 252,42	***1 245,53***	4,31
k-GH-VNS2-RND	1 245,93	1 264,07	1 254,72	5,15
k-GH-VNS3-RND	1 243,73	1 259,78	1 251,01	***3,96***
j-means-GH-VNS1	***1 236,21***	1 257,85	***1 245,97***	6,48
j-means-GH-VNS2	1 245,71	1 256,18	1 249,95	***3,09***

Moreover, calculations were carried out with various versions on some data sets for the expected number of clusters, as well as with a varying time limit allotted for one attempt to run the algorithm (Tables 2.1, 2.3, 2.6).

The results of computational experiments (Tables 2.1-2.6) showed that new search algorithms with alternating randomized neighborhoods (k-GH-VNS) have more stable results (give a lower minimum value and/or standard deviation of the objective function, a smaller spread of the achieved values) and, therefore, better performance in comparison with classical algorithms j-means and k-means (Figures 2.3-2.9) [149, 159]. At the same time, it is difficult to give an unambiguous preference to any one of the versions of the k-GH-VNS algorithm or its combined version j-means-GH-VNS. However, note that the k-GH-VNS2 combination (in which Greedy Procedure 2, i.e., Algorithm 2.3 is used first) in most cases shows worse results in comparison with other new algorithms, which casts doubt on the expediency of its application.

Table 2.2

Results of computational experiments on the mopsi-Joensuu dataset (20 clusters, 40 minutes, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
j-means	36,565	37,520	36,730	0,253
k-means	47,891	52,759	50,387	1,359
k-GH-VNS1	***36,565***	36,565	***36,565***	***0,000***
k-GH-VNS2	***36,565***	36,565	***36,565***	***0,000***
k-GH-VNS3	***36,565***	36,565	***36,565***	***0,000***
k-GH-VNS1-RND	***36,565***	36,565	***36,565***	***0,000***
k-GH-VNS2-RND	***36,565***	36,565	***36,565***	***0,000***
k-GH-VNS3-RND	***36,565***	36,565	***36,565***	***0,000***
j-means-GH-VNS1	***36,565***	36,565	***36,565***	***0,000***
j-means-GH-VNS2	***36,565***	36,565	***36,565***	***0,000***

Table 2.3

Results of computational experiments on the chess dataset (30 clusters, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
5 minutes				
j-means	8 021,22	8 102,13	8 060,24	21,05
k-means	7 989,20	8 038,81	8 019,24	13,68
k-GH-VNS1	***7 960,82***	7 978,85	7 967,27	4,99
k-GH-VNS2	***7 959,92***	7 989,01	7 974,27	8,61
k-GH-VNS3	7 998,48	8 007,96	8 003,84	***3,51***

k-GH-VNS1-RND	7 960,66	7 978,62	7 968,86	5,94
k-GH-VNS2-RND	7 961,89	7 996,10	7 972,31	8,76
k-GH-VNS3-RND	7 987,20	8 001,26	7 991,42	***3,55***
j-means-GH-VNS1	***7 958,25***	7 967,75	***7 961,82***	4,65
j-means-GH-VNS2	***7 959,03***	7 970,65	***7 963,63***	4,13
2 hours				
j-means	7 997,43	8 031,05	8 014,72	10,71
k-means	7 970,88	8 005,28	7 990,12	9,31
k-GH-VNS1	***7 958,26***	7 969,10	7 962,73	3,89
k-GH-VNS2	***7 958,25***	7 961,61	***7 959,34***	1,21
k-GH-VNS3	***7 958,26***	7 963,07	7 960,22	2,03
k-GH-VNS1-RND	***7 958,24***	7 965,03	7 960,91	1,59
k-GH-VNS2-RND	***7 958,24***	7 963,09	***7 959,57***	1,56
k-GH-VNS3-RND	***7 958,24***	7 968,36	***7 959,49***	2,64
j-means-GH-VNS1	***7 958,25***	7 958,28	***7 958,26***	***0,02***
j-means-GH-VNS2	***7 958,25***	7 960,39	***7 958,68***	***0,85***

Table 2.4

Results of computational experiments on the Europe dataset (30 clusters, 4 hours, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
j-means	7,51477E+12	7,60536E+12	7,56092E+12	29,764E+9
k-means	7,54811E+12	7,57894E+12	7,56331E+12	13,560E+9
k-GH-VNS1	***7,4918E+12***	7,49201E+12	***7,49185E+12***	***0,073E+9***
k-GH-VNS2	7,49488E+12	7,52282E+12	7,50082E+12	9,989E+9
k-GH-VNS3	***7,4918E+12***	7,51326E+12	7,49976E+12	9,459E+9
k-GH-VNS1-RND	***7,49181E+12***	7,49358E+12	7,49224E+12	0,688E+9

k-GH-VNS2-RND	***7,4918E+12***	7,51914E+12	7,49719E+12	9,776E+9
k-GH-VNS3-RND	7,49182E+12	7,51505E+12	7,4971E+12	8,159E+9
j-means-GH-VNS1	***7,4918E+12***	7,49211E+12	***7,49185E+12***	***0,112E+9***
j-means-GH-VNS2	7,49187E+12	7,51455E+12	7,4962E+12	8,213E+9

Table 2.5

Results of computational experiments on the birch3 dataset (100 clusters, 6 hours, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
j-means	3,76222E+13	3,7965E+13	3,77715E+13	0,116211E+12
k-means	7,92474E+13	8,87404E+13	8,31599E+13	3,088140E+12
k-GH-VNS1	***3,72537E+13***	3,77474E+13	3,74703E+13	0,171124E+12
k-GH-VNS2	4,21378E+13	5,01871E+13	4,52349E+13	4,333462E+12
k-GH-VNS3	***3,72525E+13***	3,74572E+13	***3,73745E+13***	0,074315E+12
k-GH-VNS1-RND	***3,72541E+13***	3,77687E+13	3,74943E+13	0,185483E+12
k-GH-VNS2-RND	3,83257E+13	4,61847E+13	4,0815E+13	2,543163E+12
k-GH-VNS3-RND	3,73131E+13	3,75242E+13	3,74164E+13	***0,061831E+12***
j-means-GH-VNS1	***3,71595E+13***	3,71807E+13	***3,71735E+13***	***0,012162E+12***
j-means-GH-VNS2	***3,72422E+13***	3,7456E+13	***3,7347E+13***	0,106977E+12

Table 2.6

Results of computational experiments on the KDDCUP04BioNormed dataset (30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
30 clusters, 500 minutes				
j-means	6 280 406	6 288 774	6 283 271	4 767,4
k-means	6 310 843	6 429 357	6 370 635	63 853,5
k-GH-VNS1	6 385 012	6 385 150	6 385 047	***51,3***
k-GH-VNS2	6 385 196	6 430 515	6 418 326	17 204,4
k-GH-VNS3	***6 267 808***	6 286 808	6 283 641	7 756,6
k-GH-VNS1-RND	6 385 016	6 385 033	6 385 023	***6,2***
k-GH-VNS2-RND	6 385 149	6 429 426	6 410 598	16 538,2
k-GH-VNS3-RND	6 386 703	6 386 808	6 386 753	***50,4***
j-means-GH-VNS1	***6 267 205***	6 267 395	***6 267 300***	134,3
j-means-GH-VNS2	***6 267 217***	6 267 421	***6 267 319***	144,2
200 clusters, 24 hours				
j-means	5 330 344	5 382 908	5 355 903	26 785,8
k-means	5 336 446	5 381 386	5 366 144	25 722,4
k-GH-VNS1	***5 294 620***	5 307 828	***5 301 224***	***9 339,5***
k-GH-VNS2	5 440 814	5 476 140	5 458 477	24 979,5
k-GH-VNS3	no result	no result		
k-GH-VNS1-RND	***5 310 067***	5 340 849	5 325 458	21 765,7
k-GH-VNS2-RND	5 368 527	5 399 695	5 384 111	22 039,6
k-GH-VNS3-RND	no result	no result		
j-means-GH-VNS1	5 430 120	5 446 222	5 438 171	11 385,8
j-means-GH-VNS2	5 500 410	5 508 985	5 504 697	***6 063,4***

Figures 2.3-2.9 show graphs of the convergence of algorithms, built by the average value of the objective function for graphical comparison of new and known algorithms for each dataset (Tables 2.1-2.6). According to the picture, abscissa is the time and the ordinate is the objective function value averaged over 30 runs.

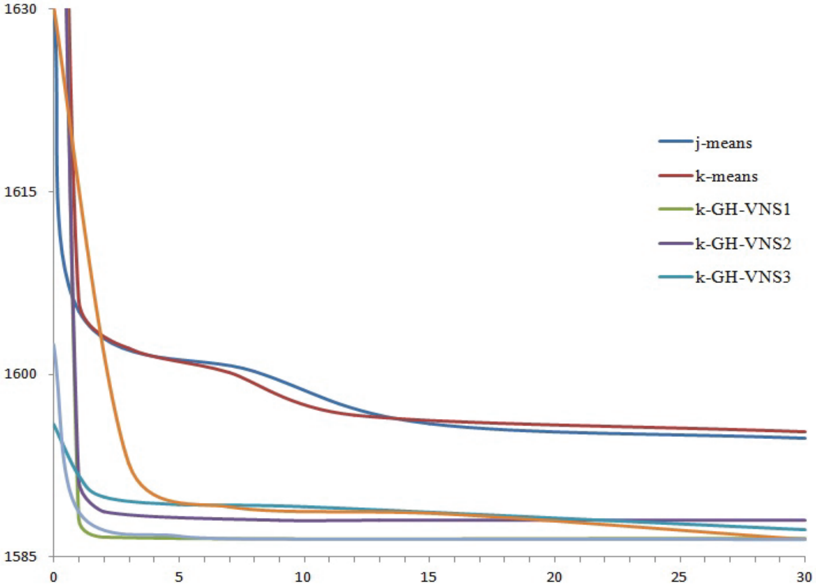


Fig. 2.3. Comparison of new and known algorithms for the ionosphere dataset (10 clusters, 30 seconds):
abscissa - time in seconds; ordinate - achieved average value of the objective function

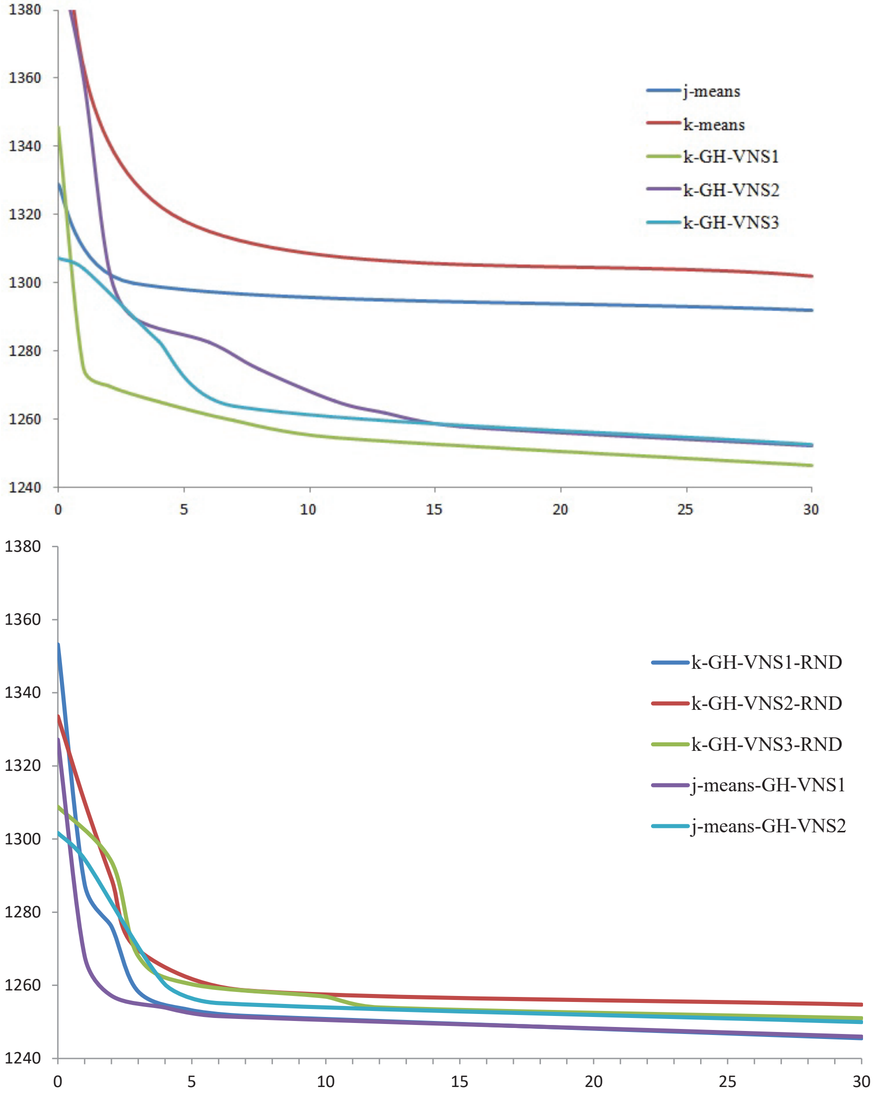


Fig. 2.4. Comparison of new and known algorithms for the ionosphere dataset (20 clusters, 30 seconds):
abscissa - time in seconds; ordinate - achieved average value of the objective function

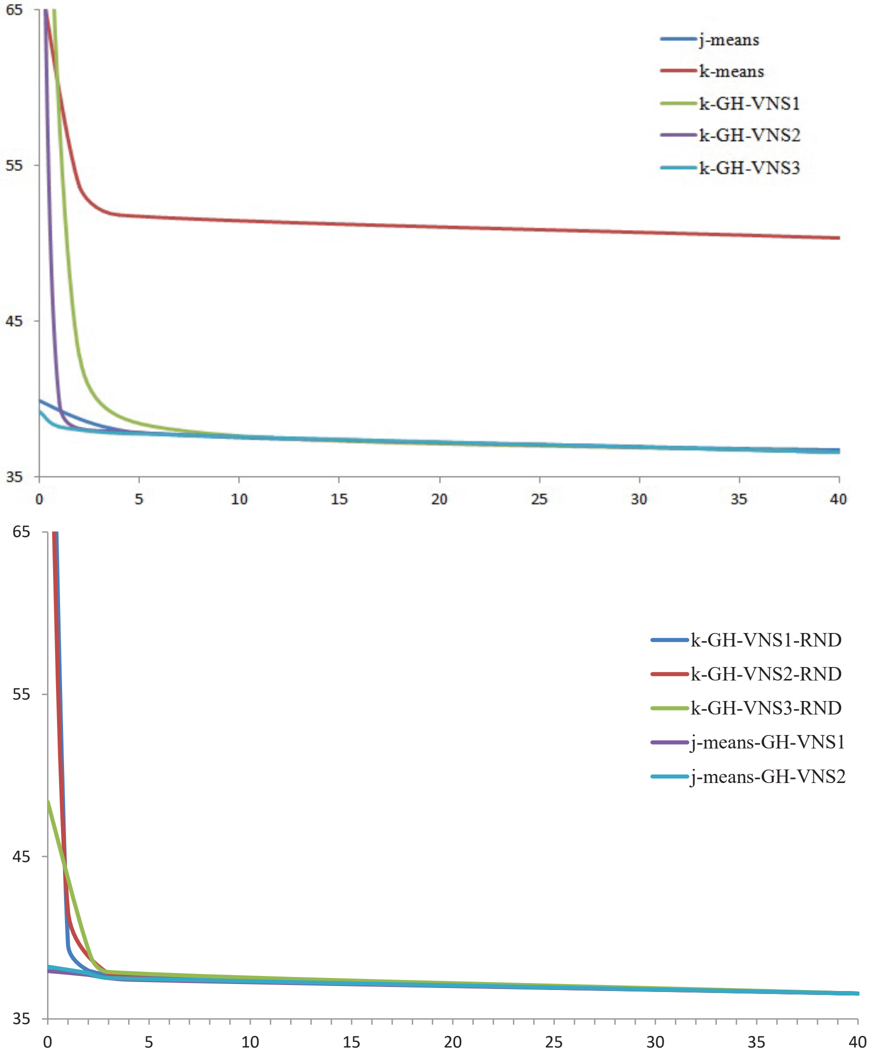


Fig. 2.5. Comparison of new and known algorithms for the mopsi-Joensuu dataset (20 clusters, 40 minutes):
abscissa - time in minutes; ordinate - achieved average value of the objective function

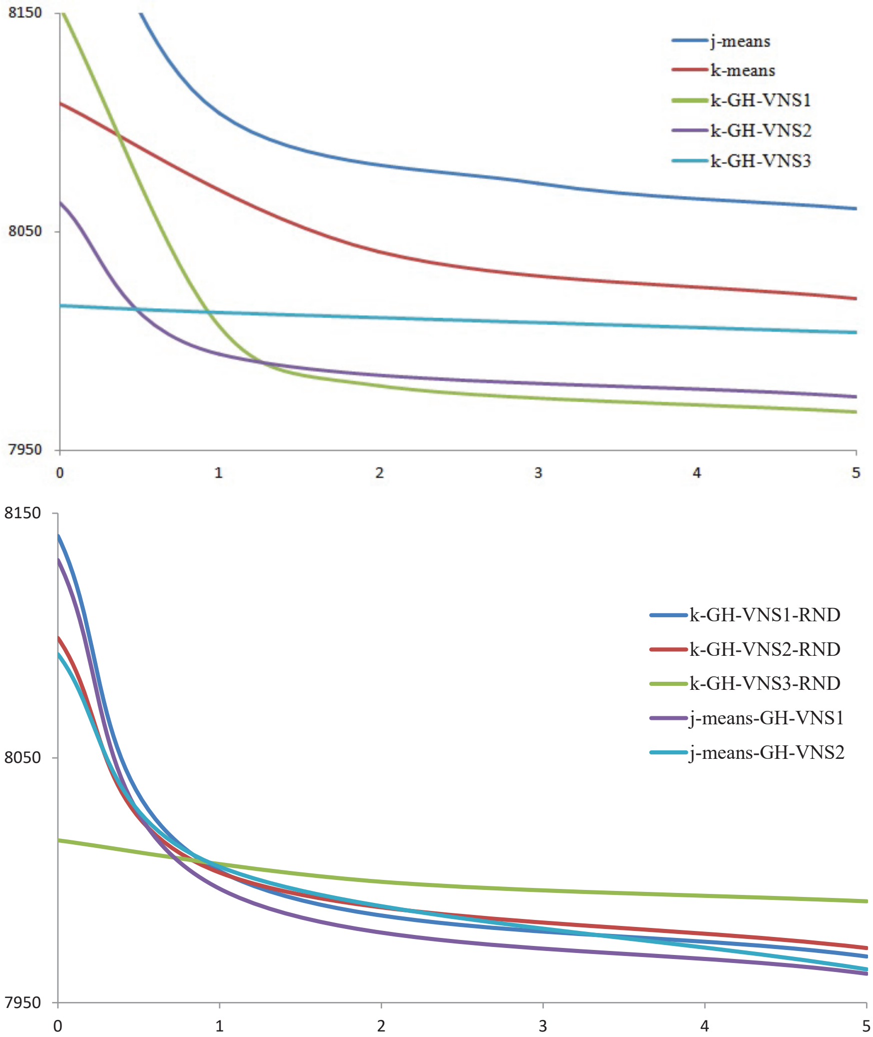
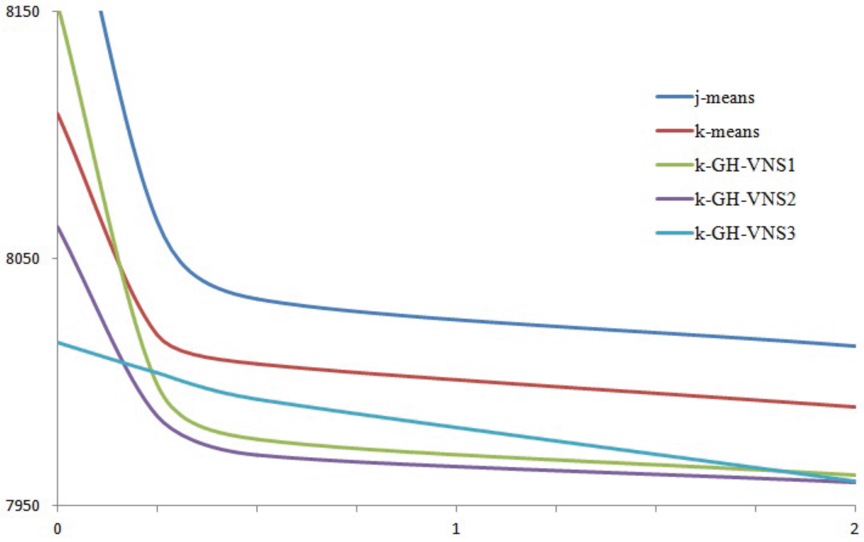


Fig. 2.6. Comparison of new and known algorithms for the chess dataset (30 clusters, 5 minutes):
abscissa - time in minutes; ordinate - achieved average value of the objective function



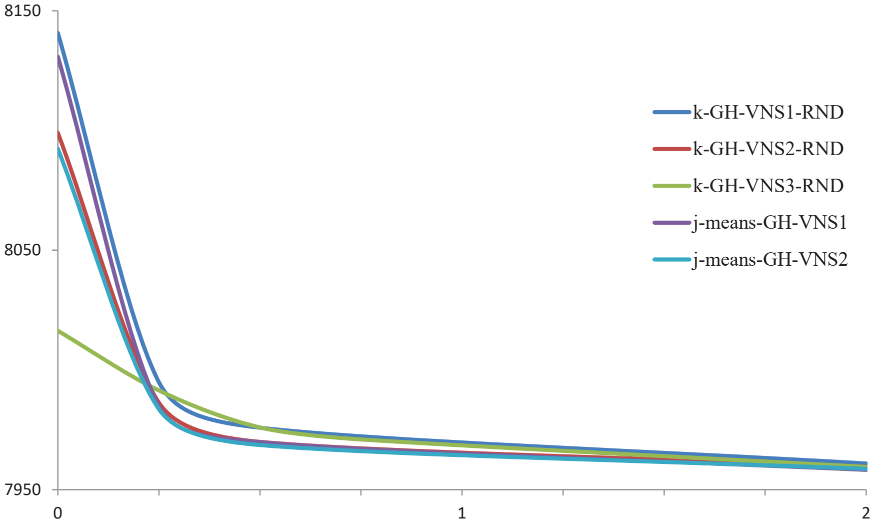


Fig. 2.7. Comparison of new and known algorithms for the chess dataset (30 clusters, 2 hours):
abscissa - time in hours; ordinate - achieved average value of the objective function

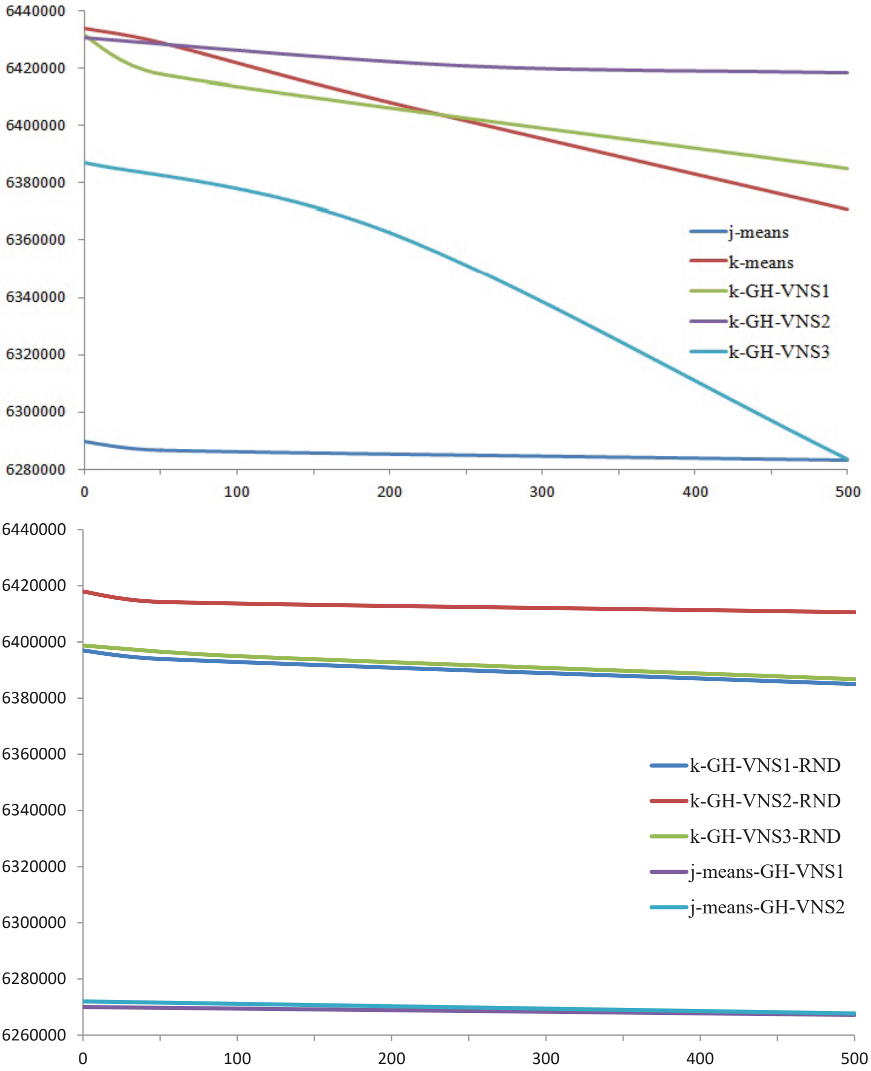


Fig. 2.8. Comparison of new and known algorithms for the KDDCUP04BioNormed dataset (30 clusters, 500 minutes): abscissa - time in minutes; ordinate - achieved average value of the objective function

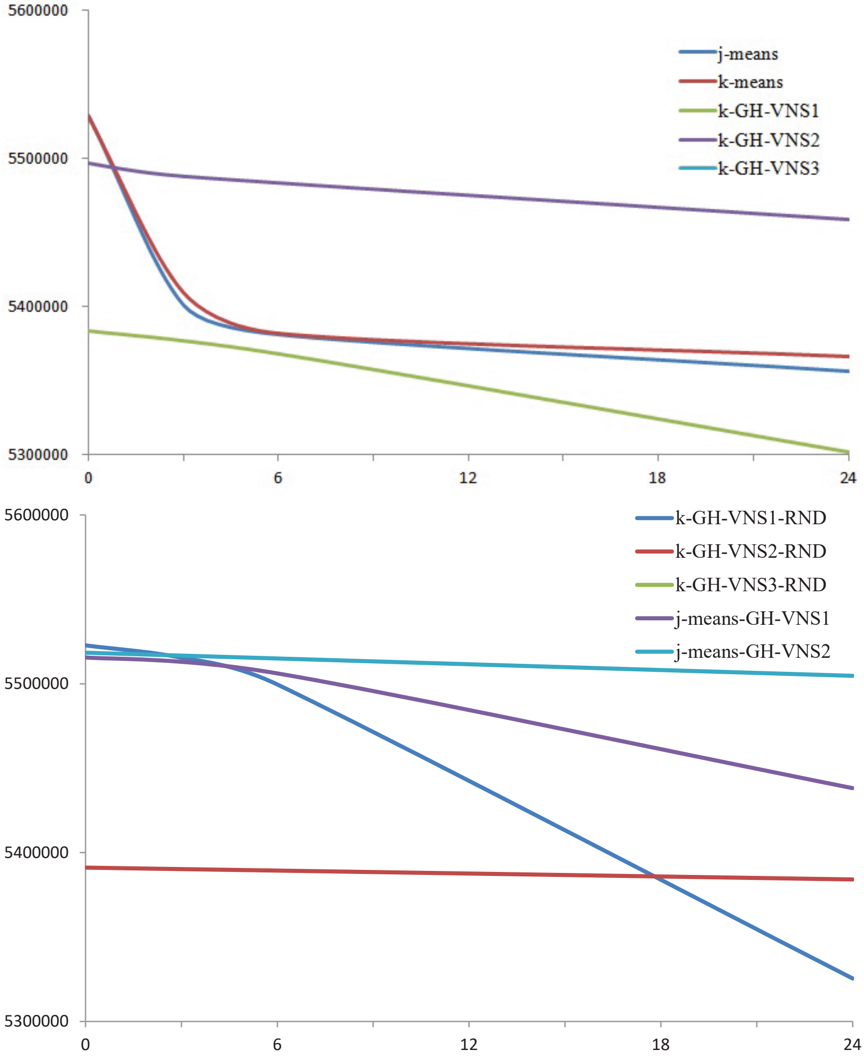


Fig. 2.9. Comparison of new and known algorithms for the KDDCUP04BioNormed dataset (200 clusters, 24 hours): abscissa - time in hours; ordinate - achieved average value of the objective function

The results of computational experiments shown in Tables 2.1-2.6 are graphically summarized in Figure 2.10.

Figure 2.10 presents the number of the best achieved values of the objective function among all solved problems (datasets from the UCI repositories and Clustering basic benchmark) for all clustering algorithms (Tables 2.1-2.6). The number of the best values of the objective function among the clustering algorithms was calculated for all computational experiments: separately for the minimum value (Min) and the root-mean-square deviation (RMSD). Also, the figure contains a column (Min + RMSD) to show which of the clustering algorithms give the best of the obtained values of the objective function simultaneously both for the best (record) value of the objective function and for the standard deviation of the achieved value of the objective function.

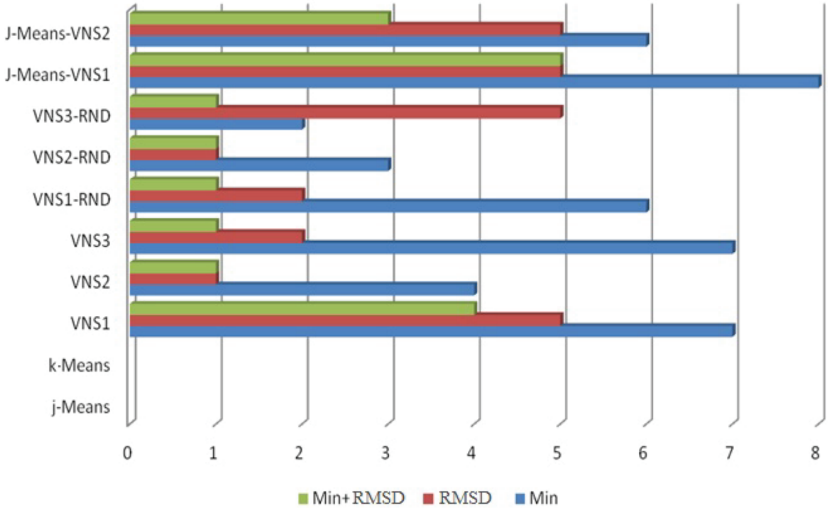


Fig. 2.10. The number of achieved the best record and the best averaged values of the objective function by each of the algorithms, calculated for all computational experiments with all datasets from the UCI and Clustering basic benchmark repositories, as well as the number of simultaneously achieved records, both in terms of the objective function value and RMSD

As test datasets, the results of non-destructive testing of prefabricated production batches of electrical radio products (ERP) were also used, carried out in a specialized test center of ITC - NPO PM JSC (Zheleznogorsk) to complete the onboard equipment of spacecraft, the composition of which is known in advance:

- 3OT122A - 2 production batches (767 data vectors, each dimension 13);
- 5514BC1T2-9A5 - 2 batches (91 data vectors, each dimension 173);
- 1526TL1 - 3 batches (1234 data vectors, each dimension 157).

The task was to divide the compiled team party into clusters corresponding to homogeneous parties, followed by an analysis of the quality of this division. The best values of the objective function (minimum value, mean value and standard deviation) are highlighted in bold italic (Tables 2.7-2.9). The graphical implementation of the convergence of algorithms based on the mean value of the objective function is shown in Figures 2.11-2.13.

According to the results of computational experiments on data sets of electrical radio products (Tables 2.7-2.9), new combined search algorithms with alternating randomized neighborhoods again gave more stable results (Figure 2.14). It should be noted that the modifications of Algorithm 2.5 (k-GH-VNS-RND), in which the number of centers in the solution S' is chosen randomly from the set $\left\{\overline{2,|S|}\right\}$, do not show themselves in the best way, and often do not give any result at all. Therefore, these algorithms are not shown in some tables in which they did not have a solution.

Table 2.7

Results of computational experiments on production batches of 3OT122A electrical radio products (2 minutes, 30 attempts)

Algorythm	Objective function value			
	Min (рекорд)	Max	Mean	Root mean square deviation
2 clusters				
j-means	***3 978,31***	3 978,31	***3 978,31***	***0,0000***
k-means	***3 978,31***	3 978,31	***3 978,31***	***0,0000***
k-GH-VNS1	***3 978,31***	3 978,31	***3 978,31***	***0,0000***

k-GH-VNS2	***3 978,31***	3 978,31	***3 978,31***	***0,0000***
k-GH-VNS3	***3 978,31***	3 978,31	***3 978,31***	***0,0000***
k-GH-VNS1-RND	***3 978,31***	3 978,31	***3 978,31***	***0,0000***
k-GH-VNS2-RND	***3 978,31***	3 978,31	***3 978,31***	***0,0000***
k-GH-VNS3-RND	***3 978,31***	3 978,31	***3 978,31***	***0,0000***
j-means-GH-VNS1	***3 978,31***	3 978,31	***3 978,31***	***0,0000***
j-means-GH-VNS2	***3 978,31***	3 978,31	***3 978,31***	***0,0000***
5 clusters				
j-means	***1 420,48***	1 420,49	***1 420,48***	0,0041
k-means	***1 420,48***	1 420,48	***1 420,48***	***0,0000***
k-GH-VNS1	***1 420,49***	1 420,50	***1 420,49***	0,0061
k-GH-VNS2	***1 420,49***	1 420,50	***1 420,49***	0,0061
k-GH-VNS3	***1 420,49***	1 420,50	***1 420,49***	0,0057
k-GH-VNS1-RND	***1 420,49***	1 420,49	***1 420,49***	***0,0000***
k-GH-VNS2-RND	***1 420,49***	1 420,50	***1 420,49***	0,0057
k-GH-VNS3-RND	***1 420,49***	1 420,50	***1 420,49***	0,0050
j-means-GH-VNS1	***1 420,49***	1 420,50	***1 420,49***	0,0061
j-means-GH-VNS2	***1 420,49***	1 420,50	***1 420,49***	0,0057
10 clusters				
j-means	***772,66***	772,70	772,68	0,0191
k-means	***772,66***	772,66	***772,66***	***0,0000***
k-GH-VNS1	***772,66***	772,66	***772,66***	***0,0000***
k-GH-VNS2	***772,66***	772,66	***772,66***	***0,0000***
k-GH-VNS3	***772,66***	772,66	***772,66***	***0,0000***
k-GH-VNS1-RND	***772,66***	772,66	***772,66***	***0,0000***
k-GH-VNS2-RND	***772,66***	772,66	***772,66***	***0,0000***
k-GH-VNS3-RND	***772,66***	772,66	***772,66***	***0,0000***
j-means-GH-VNS1	***772,66***	772,66	***772,66***	***0,0000***
j-means-GH-VNS2	***772,66***	772,66	***772,66***	***0,0000***

Table 2.8

Results of computational experiments on production batches of electrical radio products 5514BC1T2-9A5 (2 minutes, 30 attempts)

Algorythm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
2 clusters				
j-means	***10 516,87***	10 516,87	***10 516,87***	***0,0000***

k-means	***10 516,87***	10 516,87	***10 516,87***	***0,0000***
k-GH-VNS1	***10 516,87***	10 516,87	***10 516,87***	***0,0000***
k-GH-VNS2	***10 516,87***	10 516,87	***10 516,87***	***0,0000***
k-GH-VNS3	***10 516,87***	10 516,87	***10 516,87***	***0,0000***
k-GH-VNS1-RND	***10 516,87***	10 516,87	***10 516,87***	***0,0000***
k-GH-VNS2-RND	***10 516,87***	10 516,87	***10 516,87***	***0,0000***
k-GH-VNS3-RND	***10 516,87***	10 516,87	***10 516,87***	***0,0000***
j-means-GH-VNS1	***10 516,87***	10 516,87	***10 516,87***	***0,0000***
j-means-GH-VNS2	***10 516,87***	10 516,87	***10 516,87***	***0,0000***
5 clusters				
j-means	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
k-means	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
k-GH-VNS1	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
k-GH-VNS2	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
k-GH-VNS3	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
k-GH-VNS1-RND	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
k-GH-VNS2-RND	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
k-GH-VNS3-RND	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
j-means-GH-VNS1	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
j-means-GH-VNS2	***8 287,83***	8 287,83	***8 287,83***	***0,0000***
10 clusters				
j-means	7 060,45	7 085,67	7 073,55	8,5951
k-means	7 046,33	7 070,83	7 060,11	8,8727
k-GH-VNS1	***7 001,12***	7 009,53	7 004,48	4,3453
k-GH-VNS2	***7 001,12***	7 010,59	***7 002,26***	***2,9880***
k-GH-VNS3	***7 001,12***	7 009,53	7 003,01	***3,1694***
j-means-GH-VNS1	***7 001,12***	7 001,12	***7 001,12***	***0,0000***
j-means-GH-VNS2	***7 001,12***	7 011,94	7 003,88	4,4990

Table 2.9

Results of computational experiments on production batches of 1526TL1 electrical and radio products (2 minutes, 30 attempts)

Algorythm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
3 clusters				
j-means	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
k-means	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
k-GH-VNS1	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
k-GH-VNS2	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
k-GH-VNS3	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
k-GH-VNS1-RND	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
k-GH-VNS2-RND	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
k-GH-VNS3-RND	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
j-means-GH-VNS1	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
j-means-GH-VNS2	***86 599,77***	86 599,77	***86 599,77***	***0,0000***
5 clusters				
j-means	***63 337,29***	63 337,56	***63 337,46***	0,1211
k-means	***63 337,29***	63 337,29	***63 337,29***	***0,0000***
k-GH-VNS1	***63 337,47***	63 337,56	***63 337,55***	0,0280
k-GH-VNS2	***63 337,56***	63 337,56	***63 337,56***	***0,0000***
k-GH-VNS3	***63 337,56***	63 337,56	***63 337,56***	***0,0000***
k-GH-VNS1-RND	***63 337,56***	63 337,56	***63 337,56***	***0,0000***
k-GH-VNS2-RND	***63 337,56***	63 337,56	***63 337,56***	***0,0000***
k-GH-VNS3-RND	***63 337,56***	63 337,56	***63 337,56***	***0,0000***
j-means-GH-VNS1	***63 337,56***	63 337,56	***63 337,56***	***0,0000***
j-means-GH-VNS2	***63 337,56***	63 337,56	***63 337,56***	***0,0000***
10 clusters				
j-means	***43 841,97***	43 843,51	***43 842,59***	0,4487
k- means	***43 842,10***	43 844,66	43 843,38	0,8346
k-GH-VNS1	***43 841,97***	43 844,18	***43 842,34***	0,9000
k-GH-VNS2	***43 841,97***	43 844,18	43 843,46	1,0817
k-GH-VNS3	***43 841,97***	43 842,10	***43 841,99***	***0,0424***
j-means-GH-VNS1	***43 841,97***	43 841,97	***43 841,97***	***0,0000***
j-means-GH-VNS2	***43 841,97***	43 844,18	***43 842,19***	***0,6971***

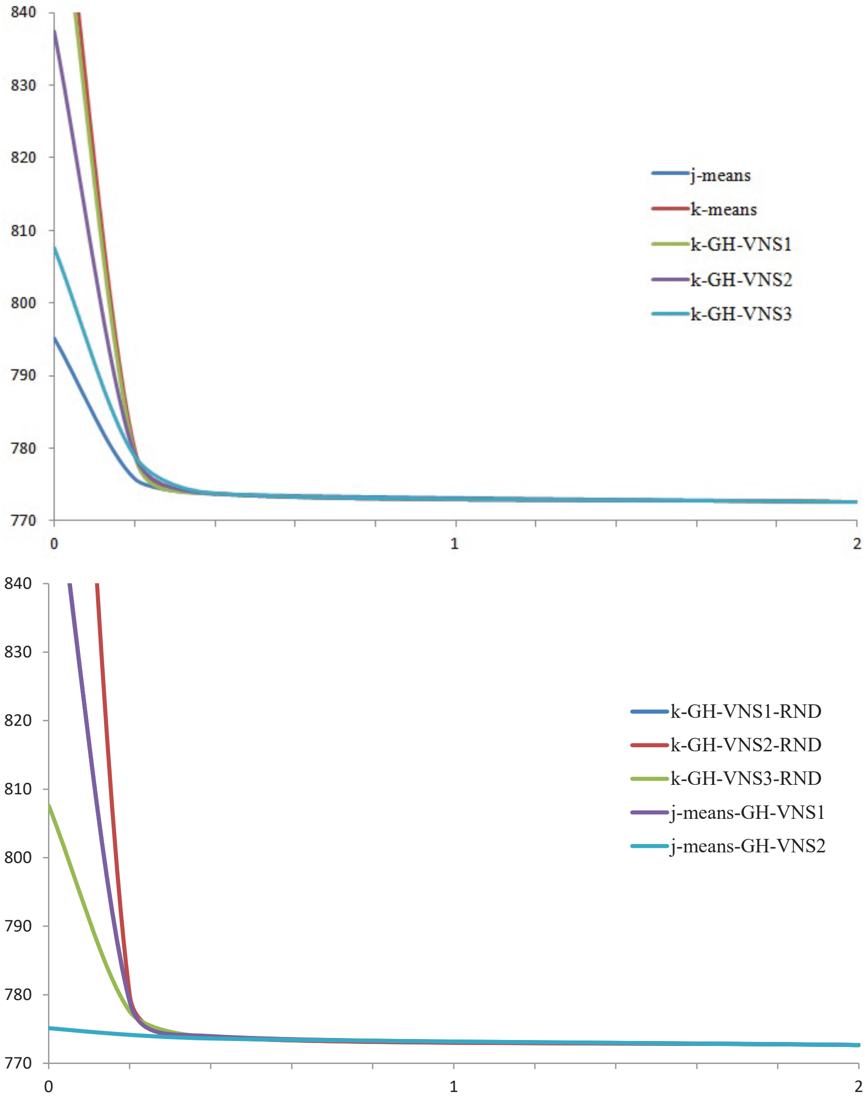


Fig. 2.11. Comparison of new and known algorithms for data set 3OT122A (10 clusters, 2 minutes): along the abscissa axis - time in minutes; along the ordinate axis - the achieved average value of the objective function

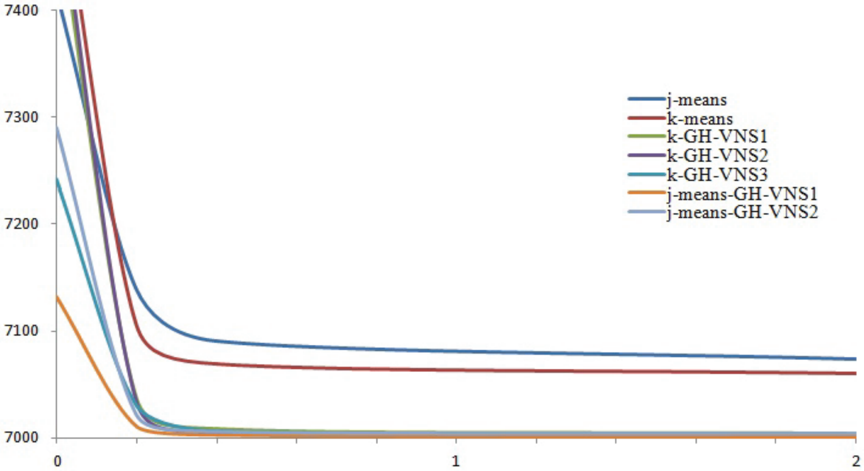


Fig. 2.12. Comparison of new and known algorithms for data set 5514BC1T2-9A5 (10 clusters, 2 minutes): along the abscissa axis - time in minutes; along the ordinate axis - the achieved average value of the objective function

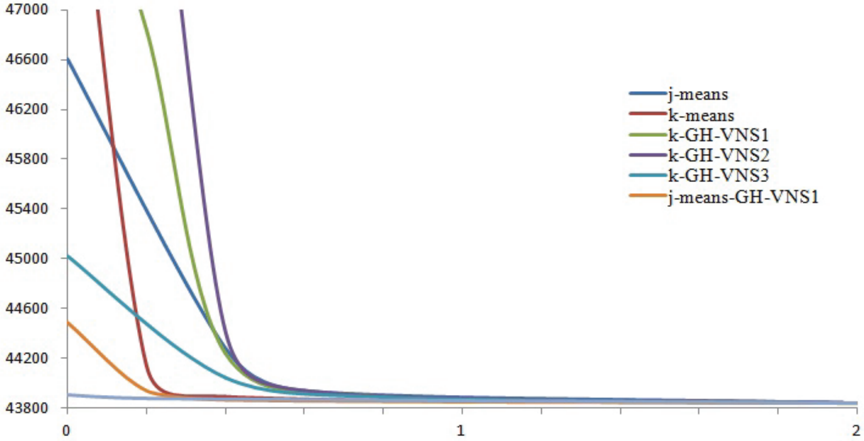


Fig. 2.13. Comparison of new and known algorithms for data set 1526TL1 (10 clusters, 2 minutes): along the abscissa axis - time in minutes; along the ordinate axis - the achieved average value of the objective function

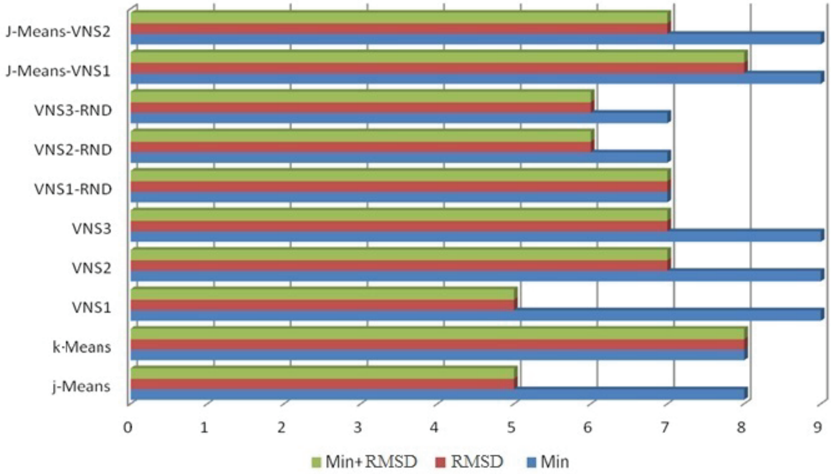


Fig. 2.14. Number of achieved best records and best average values of the objective function by each of the algorithms, calculated for all computational experiments with all data sets of test results for electrical and radio products, as well as the number of simultaneously achieved records, both in terms of the value of the objective function and in terms of SKO

The results of earlier computational experiments on data sets of electrical radio products with various modifications of the genetic algorithm were applied for a more complete comparison of computational experiments results of new algorithms with the known algorithms [142]. The comparative results are given in the Appendix. data sets were applied for calculations. They are the test results of prefabricated batches of electrical and radio products:

- 1526TL1 - 3 batches (1234 data vectors, each dimension 157);
- 2Д522Б - 5 batches (3711 data vectors, each dimension 10);
- H5503XM1 - 5 batches (3711 data vectors, each 229 dimensions).

According to the Appendix, the values of the objective function obtained by new algorithms in some cases turn out to be significantly better (according to the achieved value of the objective function) and more stable than the results of the genetic algorithm. Nevertheless, the authors state the competitiveness of new algorithms both in comparison with the classical algorithms of k-means and j-means, and with genetic algorithms, including algorithms of the greedy heuristic method, as well as with deterministic algorithms.

The next section presents the implementation of greedy heuristic algorithms using the CUDA architecture and the study of their properties when solving large-scale problems.

2.4. Implementation of greedy heuristic clustering algorithms for massively parallel systems

CUDA (the abbreviation for Compute Unified Device Architecture) is a parallel computing platform and programming model specially developed by NVIDIA for general computing on graphics processors (GPUs). They can can significantly increase computing performance [160]. A lot of new solutions and initiatives in various research fields are being implemented with CUDA such as video and image processing, computational biology and chemistry, fluid dynamics modeling, seismic analysis, image reconstruction (obtained by computed tomography), ray tracing and much more.

Algorithms that apply parallel data processing (when the same sequence of mathematical operations is applied to a large amount of data) give excellent results when calculating on the GPU, especially if the algorithm is in principle well parallelized and the ratio of the number of arithmetic instructions to the number of calls to memory. Moreover, a large volume of data and high density of mathematical operations eliminate the need for large caches, as on the central processing unit (CPU).

The separation of vertex and fragment shader processors in hardware has disappeared due to the unified model of shaders (programs for the GPU). Shader processors can now be configured to perform both tasks depending on the requirements of the application. Moreover, a special type of shader, the geometry shader, has been introduced. It helps generate geometric elements in hardware on a computer [161]. Starting with the G80 family of GPUs, NVIDIA has supported this new shader model, which is a departure from previous GPU designs. The GPU now consists of so-called multiprocessor systems that host a number of stream processors that are ideal for massively parallel computing.

The GPU is viewed as a set of multiprocessors executing parallel threads (Figure 2.15). Threads are grouped into data blocks and execute the same instructions on different data in parallel. One or more blocks are directly connected to the hardware multiprocessor. Here, the timing determines the order of execution. Within a single block, threads can be synchronized at any point in execution. A definite execution of a blocking order is not guaranteed. Further blocks are grouped into networks. Communication and synchronization across blocks is not possible, execution on the order of blocks with a network is not defined. Threads and blocks can be organized in three and two dimensions, respectively. A thread is assigned an identifier depending on its position in the block; a block is also assigned an identifier depending on its position in the grid. The thread and thread block ID are available at runtime, allowing you to specify specific memory access patterns based on your chosen layouts. Each thread in the GPU executes the same procedure, known as a kernel [160, 162].

Threads have access to different kinds of memory. Each thread is written to a local register very quickly, and local memory is assigned to it. All threads have access to a block of local shared memory within a single block. It can be accessed as quickly as registers, depending on access patterns. Registers, local memory and shared memory are resource limited. Portions of device memory can be used as texture or persistent memory that benefit from on-chip caching. Persistent memory is optimized for read-only operations, texture memory is optimized for specific access patterns. Threads also have access to uncached general purpose memory or global memory [160].

Various bugs can degrade GPU performance. First, sharing memory between multiple parallel threads can lead to so-called bank conflicts that serialize the execution of these threads and therefore reduce parallelism. Secondly, when accessing global memory addresses, it must be a multiple of 4, 8, or 16, otherwise the access can be compiled for several instructions and, therefore, accesses. Moreover, addresses accessed simultaneously by multiple threads in global memory must be located so that memory accesses can be combined into a single, contiguous, aligned memory access. This is often referred to as memory pooling. Another factor is the so-called occupancy. Fillability determines how many blocks, and therefore threads, are actually running in parallel. Since shared memory and registers are limited resources, the GPU can only execute a certain number of blocks in parallel. Therefore, it is imperative to optimize the use of shared memory and register as many parallel blocks and threads as possible [160, 162].

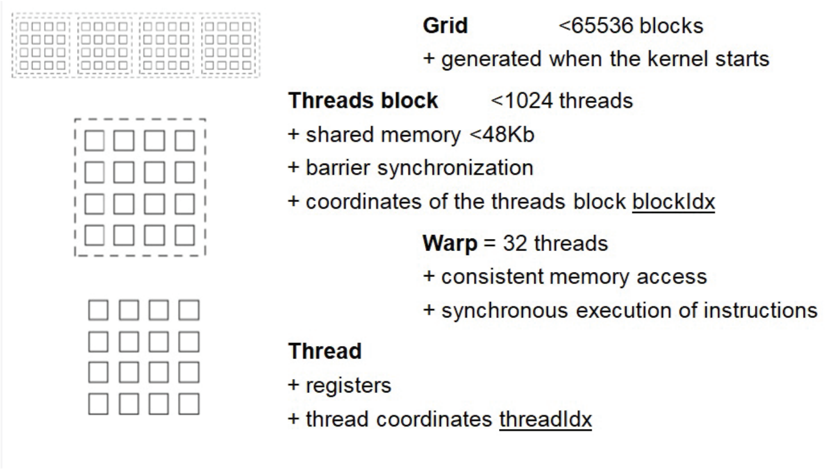

Grid <65536 blocks
+ generated when the kernel starts
Threads block <1024 threads
+ shared memory <48Kb
+ barrier synchronization
+ coordinates of the threads block blockIdx
Warp = 32 threads
+ consistent memory access
+ synchronous execution of instructions
Thread
+ registers
+ thread coordinates threadIdx

Fig. 2.15. Model CUDA

Currently, many parallel algorithms adapted to the CUDA architecture have been implemented on GPUs [162-165].

As a rule, each thread when applying the CUDA architecture performs very simple operations for processing information associated with the simplest object. Since the architecture was originally developed for image processing, such an object is a pixel. Such an object is either data objects or clusters, represented, for example, by centroid coordinates in various parts of automatic grouping algorithms. Thus, each computation thread is an intelligent agent responsible for processing data associated with its elementary object - a data object or a cluster center (for the k-means model). The steps of the k-means algorithm, which are responsible for partitioning a set of data objects into clusters, represented by the coordinates of their centroids, alternate with the steps associated with recalculating the cluster centroids. In the first case, when implemented for the CUDA architecture, threads can be launched for each of the data objects ("an agent" associated with the data object must determine which centroid it is closest to and "assign" itself to the corresponding cluster). In the second case, the centroid agents, using the data of the objects "assigned" to them, recalculate their coordinates, the flow is launched for each centroid.

Consider a parallel implementation of the k-means algorithm (Algorithm 1.1) for the CUDA architecture on the GPU [162, 165, 166].

1.1. The authors applied the following implementation of Step 1 of Algorithm 1.1.

The authors applied one computation thread (virtually without parallelization) for the first part of the parallel algorithm. It implements the 1st step of Algorithm 1.1.

Algorithm 2.8 CUDA implementation of step 1 of Algorithm 1.1, part 1

```
$X'_j$=0 for all $j \in \{\overline{1,k}\}$. // Here, $X'_j$ are the vectors applied to calculate the new cluster centers/centroids.
$counter_j$=0 for all $j \in \{\overline{1,k}\}$. // object counters for each cluster
```

The authors applied Ntrreads = 512 threads for each CUDA block for the second part of the algorithm. It implements the 1st step of Algorithm 1.1. The number of blocks is calculated as follows:

$$N_{blocks}=(N+N_{threads}-1)/N_{threads}. \tag{2.1}$$

Thus, each thread processes only one object (vector) of data.

Algorithm 2.9 CUDA implementation of step 1 of Algorithm 1.1, part 2

```
$i = blockIdx.x \times blockDim.x + threadIdx.x$ .
If $i>N$ then return.
$j'$=arg $\min_j \left\| A_j - X_i \right\|^2$ . // cluster number
$X'_{j'}=X'_{j'}+A_i$.
$C_i=j'$. // assign $A_i$ for cluster $j'$. Ai j'.
$counter_{j'}=counter_{j'}+1$.
Synchronize threads.
```

The authors applied $N_{trreads}$ = 512 for the part of the algorithm that implements the 2nd step of Algorithm 1.1. The number of blocks is calculated as $N_{blocks2} = (k + N_{threads}-1) / N_{threads}$. we used Ntrreads = 512 threads for each CUDA block.

Algorithm 2.10 CUDA implementation of step 2 of Algorithm 1.1

```
$j$ = blockIdx.x * blockDim.x + threadIdx.x .
If $j>k$ then return.
$X_{j'} = X'_i/counter_j$.
Synchronize threads.
```

The performance of the k-means algorithm with large amounts of data is a problem, especially when finding the correct parameter k can only be done by running several runs with different numbers of clusters. Moreover, Algorithm 2.1 assumes multiple runs of the k-means algorithm (or other local search method), and the number of these runs increases with the number of clusters (quadratic dependence). The authors apply a GPU-optimized strategy for k-means, as well as a procedure adapted to the CUDA architecture to exclude clusters from the solution. It is a mandatory and most computationally expensive step in the greedy agglomerative heuristic procedure [166, 167]. The authors implemented step 2 of Algorithm 2.1 on the graphics processing unit (GPU). At this step, Algorithm 2.1 calculates the total distance after removing one cluster: $F'_{i'}$ = *F(S')*, where $S' = S \setminus \{X_{i'}\}$. Having calculated F(S), we can calculate $F'_{i'}$=*F*(*S'*) = *F*(*S*) + $\sum_{l=1}^{N} \Delta D_l$.

$$\Delta D_l . = \begin{cases} 0, & C_{i'} \neq l, \\ \left(\min_{j \in \{\overline{1,k}\}, j \neq i'} \left\| A_j - X_j \right\|^2 \right) - \left\| A_j - X_{C_{i'}} \right\|^2, & C_{i'} = l. \end{cases} \tag{2.2}$$

where *l* is the cluster number. Here we used 512 threads (the number is chosen experimentally) for each CUDA block. The number of blocks is calculated in accordance with (2.1). First, the variable *sumD* is initialized with a value of 0. Then, for each data vector, the following algorithm is run and calculated (Figure 2.16).

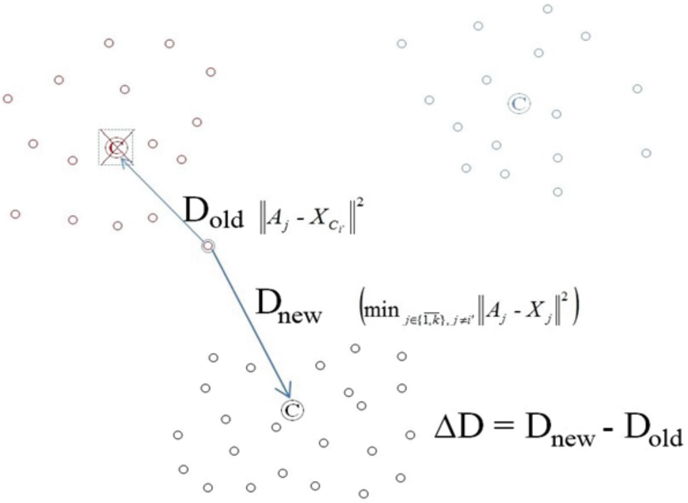

C
C
D_{old} $\|A_j - X_{c_{i'}}\|^2$
D_{new} $(\min_{j\in\{\overline{1,k}\},\, j\neq i'} \|A_j - X_j\|^2)$
C
$\Delta D = D_{new} - D_{old}$

Fig. 2.16. Calculation of the distance increment ΔD when the centroid is removed

Algorithm 2.11 CUDA implementation of step 2 of Algorithm 2.1

```
$l = blockIdx.x \times blockDim.x + threadIdx.x$ .
If $l > k$ then return.
Estimate $\Delta D_l$ in accordance with (2.2).
If $\Delta D_l > 0$ then atomicAdd ($sumD$, $\Delta D_l$).
Synchronize threads.
```

Thus, each thread of Algorithm 2.11 performs the function of an intelligent agent that determines the contribution of each data vector to the increment of the objective function after the removal of the l-th cluster.

All other algorithms run on the central processing unit (CPU).

Thus, parallel algorithms with a greedy agglomerative heuristic procedure were proposed for solving problems of automatic grouping of a large amount of data, adapted to the CUDA architecture, in which each thread plays the role of an intelligent agent associated with a specific data vector and responding to such events, like changing coordinates or deleting centroids.

2.5. Results analysis of of computational experiments for massively parallel systems

For the study, as before, the authors used classical data sets from the UCI and Clustering basic benchmark repositories [155, 156].

The test system consisted of an Intel Core 2 Duo E8400CPU, 4GBRAM. NVIDIA GeForce 9600 GT GPU, with 2048 MB RAM.

It should be noted that all the algorithms used for these computational calculations were implemented for parallel computing on a GPU (CUDA implementation). Therefore, to avoid confusion with the algorithms mentioned earlier in this chapter, the index "G" (k-средних^G, k-GH-VNS1^G, k-GH-VNS2^G, k-GH-VNS3^G, k-GH-VNS1-RND^G, k-GH-VNS2-RND^G, k-GH-VNS3-RND^G, j-means^G, j-means-GH-VNS1^G, j-means-GH-VNS2^G, GA-FULL^G, GA-MIX^G, GA-ONE^G). For all data sets, 30 attempts were made to run each of the 13 algorithms.

Only the best results achieved in each attempt were recorded. Then minimum and maximum values (Min, Max), mean value (Mean) and standard deviation (RMS) were calculated from these results for each algorithm. Algorithms j-means and k-means were run in multistart mode.

The best objective function values (minimum value, mean value and standard deviation) are in bold italics (Tables 2.10-2.13).

Table 2.10

Results of computational experiments on the Mopsi-Joensuu dataset (180 seconds, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
100 clusters				
k- means ^{G}	20,2234	25,1256	22,6732	1,9230
k-GH-VNS1^G	1,8518	2,0704	1,9320	0,0996
k-GH-VNS2^G	***1,6519***	1,7969	1,7335	0,0504
k-GH-VNS3^G	1,6745	1,7950	1,7301	***0,0444***
k-GH-VNS1-RND^G	1,9142	2,9365	2,2084	0,3680
k-GH-VNS2-RND^G	1,7589	2,0456	1,8427	0,1026
k-GH-VNS3-RND^G	1,6558	1,8107	1,7204	0,0646
j-means^G	1,8600	10,2344	4,0787	3,4959
j-means-GH-VNS1^G	1,7801	2,1694	1,9197	0,1543

j-means-GH-VNS2^G	1,7337	2,0676	1,9031	0,1471
GA-FULL^G	***1,6544***	1,7569	***1,6760***	***0,0398***
GA-MIX^G	1,6600	17,7807	5,4884	6,5581
GA-ONE^G	19,0837	33,0772	26,8381	4,5549
300 clusters				
k- means ^{G}	5,6141	8,9812	7,7135	1,1162
k-GH-VNS1^G	2,0335	3,4027	2,6656	0,4973
k-GH-VNS2^G	5,1070	11,1468	8,9344	2,2980
k-GH-VNS3^G	***0,1432***	0,2974	***0,1836***	***0,0582***
k-GH-VNS1-RND^G	2,2020	4,3911	2,7338	0,8446
k-GH-VNS2-RND^G	6,7474	14,6131	10,9959	2,6691
k-GH-VNS3-RND^G	***0,1533***	14,4612	9,1619	5,6364
j-means^G	2,3443	7,1081	4,1037	1,7900
j-means-GH-VNS1^G	2,4097	12,8224	9,9201	3,9420
j-means-GH-VNS2^G	3,7229	6,9412	5,4652	1,3822
GA-FULL^G	0,2073	3,6894	1,2855	1,5409
GA-MIX^G	0,7039	2,5733	1,4348	0,6968
GA-ONE^G	8,0874	15,9837	11,8232	3,1623

Table 2.11

Results of computational experiments on the BIRCH3 dataset (100 clusters, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
60 seconds				
k- means ^{G}	8,18676E+13	9,96542E+13	8,98255E+13	8,37212E+12
k-GH-VNS1^G	3,71973E+13	3,76732E+13	3,73639E+13	0,18509E+12
k-GH-VNS2^G	3,73240E+13	4,06161E+13	3,91485E+13	1,14305E+12
k-GH-VNS3^G	3,72082E+13	3,72550E+13	***3,72422E+13***	***0,01998E+12***
k-GH-VNS1-RND^G	3,71993E+13	3,76607E+13	3,73757E+13	0,18322E+12
k-GH-VNS2-RND^G	3,98574E+13	5,17877E+13	4,47900E+13	4,74952E+12
k-GH-VNS3-RND^G	***3,71558E+13***	3,73328E+13	***3,72362E+13***	0,06507E+12
j-means^G	5,30805E+13	13,2286E+13	7,91183E+13	28,2000E+12
j-means-GH-VNS1^G	no result	no result		
j-means-GH-VNS2^G	no result	no result		

GA-FULL^G	3,74076E+13	3,84774E+13	3,75950E+13	0,34167E+12
GA-MIX^G	3,76402E+13	4,13519E+13	3,84577E+13	1,44968E+12
GA-ONE^G	6,36816E+13	9,10870E+13	7,47659E+13	11,6766E+12
600 seconds				
k- means ^{G}	7,98405E+13	9,96542E+13	8,93187E+13	9,04845E+12
k-GH-VNS1^G	***3,71474E+13***	3,71933E+13	***3,71778E+13***	***0,02348E+12***
k-GH-VNS2^G	***3,71474E+13***	3,72261E+13	***3,71834E+13***	0,02595E+12
k-GH-VNS3^G	***3,71473E+13***	3,72453E+13	***3,71817E+13***	0,03723E+12
k-GH-VNS1-RND^G	***3,71474E+13***	3,71932E+13	***3,71775E+13***	***0,02326E+12***
k-GH-VNS2-RND^G	***3,71474E+13***	3,72275E+13	***3,71853E+13***	0,03177E+12
k-GH-VNS3-RND^G	***3,71474E+13***	3,72275E+13	***3,71857E+13***	0,03163E+12
j-means^G	4,03266E+13	4,5392E+13	4,23065E+13	1,77787E+12
j-means-GH-VNS1^G	no result	no result		
j-means-GH-VNS2^G	no result	no result		
GA-FULL^G	3,72332E+13	3,74141E+13	3,72741E+13	0,06510E+12
GA-MIX^G	3,71525E+13	3,72071E+13	3,71949E+13	***0,02097E+12***
GA-ONE^G	3,71495E+13	3,7233E+13	3,71906E+13	0,04180E+12

Table 2.12

Results of computational experiments on the chess dataset (50 clusters, 120 seconds, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
k- means ^{G}	6 926,22	6 958,36	6 941,13	11,2781
k-GH-VNS1^G	***6 851,11***	6 855,66	***6 853,08***	***1,5482***
k-GH-VNS2^G	***6 851,07***	6 857,08	***6 853,96***	2,4912
k-GH-VNS3^G	***6 851,15***	6 859,06	6 854,82	3,5286
k-GH-VNS1-RND^G	***6 851,29***	6 859,57	***6 853,93***	2,9739
k-GH-VNS2-RND^G	***6 851,25***	6 861,01	6 857,15	3,6077
k-GH-VNS3-RND^G	***6 851,30***	6 855,86	***6 853,93***	1,7071
j-means^G	6 938,97	6 987,53	6 962,71	18,6573
j-means-GH-VNS1^G	6 931,44	6 994,70	6 963,11	23,9752
j-means-GH-VNS2^G	6 962,87	6 994,55	6 980,95	10,77
GA-FULL^G	6 864,33	6 867,14	6 865,68	***1,2282***
GA-MIX^G	***6 851,41***	6 858,32	6 854,64	2,9540
GA-ONE^G	***6 851,44***	6 860,75	6 856,01	4,0019

Table 2.13

Results of computational experiments on the data set KDDCUP04BioNormed (2000 clusters, 14 hours, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
k- means ^{G}	4 424 475	4 426 251	4 425 137	***786,5***
k-GH-VNS1^{G}	4 358 583	4 386 584	4 367 311	12 966,3
k-GH-VNS2^{G}	4 338 584	4 419 181	4 378 916	42 724,9
k-GH-VNS3^{G}	***4 311 992***	4 318 547	***4 315 658***	2 721,5
k-GH-VNS1-RND^{G}	no result	no result		
k-GH-VNS2-RND^{G}	no result	no result		
k-GH-VNS3-RND^{G}	no result	no result		
j-means^{G}	4 390 323	4 404 301	4 396 180	5 912,2
j-means-GH-VNS1^{G}	no result	no result		
j-means-GH-VNS2^{G}	no result	no result		
GA-FULL^{G}	***4 314 647***	4 319 851	***4 316 581***	2 847,4
GA-MIX^{G}	4 332 422	4 354 462	4 342 210	11 224,5
GA-ONE^{G}	4 426 306	4 431 211	4 428 233	2 615,5

According to the results of computational experiments (Tables 2.10-2.13), it can be seen that the new k-GH-VNS3G algorithm (using a greedy procedure with full set union) consistently shows the best results on all data sets when implemented in parallel on a GPU and with sufficient running time, since this heuristic, as a rule, at the very first iterations fall into the region of sufficiently “good” values of the objective function. At the same time, this heuristic often "gets stuck" in this area, and improves the known solution step by step, while other variants of the greedy heuristic procedure are able to improve the existing solution step by step. Therefore, it is very important which of the three heuristics the search starts with (the O_{start} parameter, which specifies the neighborhood number in the GH-VNS^{G} algorithm). When time is scarce, other variants of the GH-VNS^{G} algorithm take precedence.

Table 2.14

Comparison of the results of the algorithms on the CPU and GPU for the birch3 dataset (100 clusters, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
GPU 1 minute				
k- means ^{G}	8,18676E+13	9,96542E+13	8,98255E+13	8,37212E+12
j-means^{G}	5,30805E+13	13,2286E+13	7,91183E+13	28,2000E+12
k-GH-VNS1^{G}	***3,71973E+13***	3,76732E+13	3,73639E+13	0,18509E+12
k-GH-VNS2^{G}	3,73240E+13	4,06161E+13	3,91485E+13	1,14305E+12
k-GH-VNS3^{G}	3,72082E+13	3,72550E+13	***3,72422E+13***	***0,01998E+12***
GPU 10 minutes				
k- means ^{G}	7,98405E+13	9,96542E+13	8,93187E+13	9,04845E+12
j-means^{G}	4,03266E+13	4,5392E+13	4,23065E+13	1,77787E+12
k-GH-VNS1^{G}	***3,71474E+13***	3,71933E+13	***3,71778E+13***	***0,02348E+12***
k-GH-VNS2^{G}	***3,71474E+13***	3,72261E+13	3,71834E+13	0,02595E+12
k-GH-VNS3^{G}	***3,71473E+13***	3,72453E+13	3,71817E+13	0,03723E+12
CPU 6 hours				
k- means	7,92474E+13	8,87404E+13	8,31599E+13	3,088140E+12
j-means	3,76222E+13	3,7965E+13	3,77715E+13	0,116211E+12
k-GH-VNS1	3,72537E+13	3,77474E+13	3,74703E+13	0,171124E+12
k-GH-VNS2	4,21378E+13	5,01871E+13	4,52349E+13	4,333462E+12
k-GH-VNS3	***3,72525E+13***	3,74572E+13	***3,73745E+13***	***0,074315E+12***

In studies connected with the BIRCH3 data set without the use of CUDA technology (Table 2.5), the best minimum objective function value of 3.72525E+13 was obtained under the condition of 6 hours for each attempt. When calculating using a graphics processor (Table 2.11), the minimum value of the objective function 3.71473E + 13 was obtained by the same algorithm, but in 10 minutes and 3.71973E + 13 in 1 minute. According to Figure 2.17, the result when using a graphics accelerator turned out to be more accurate (Table 2.14), and the time spent was several orders of magnitude less (in this case, 360 times).

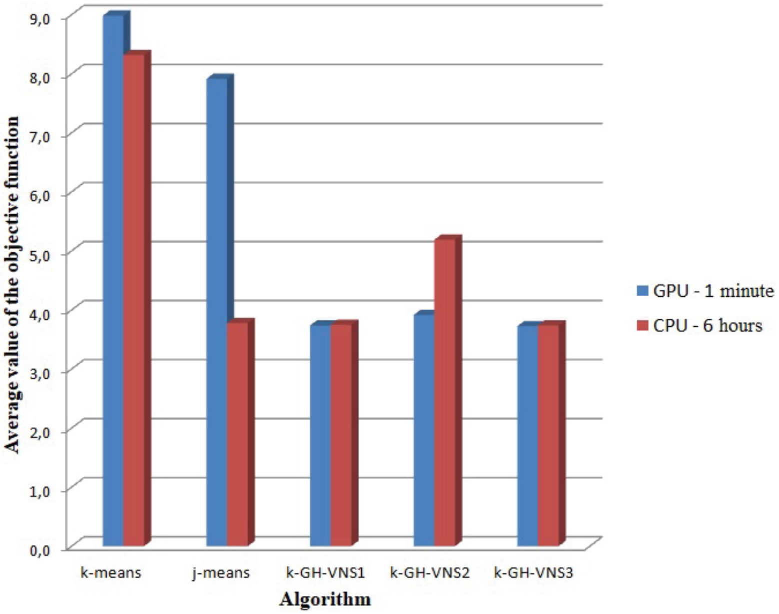


Fig. 2.17. Comparison of the results of the algorithms on the CPU and GPU for the data set birch3

Also, as test data sets, the results of non-destructive tests of prefabricated production batches of electrical and radio products were also used. They all were conducted in a specialized test center for completing the onboard equipment of spacecraft, the composition of which is known in advance (Table 2.15). The best objective function values (minimum value, mean value, and standard deviation) are in bold italics.

Table 2.15

Results of computational experiments on data set 1526IE10 (10 clusters, 20 seconds, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
k- means ^{G}	3 925,08	3 925,09	3 925,09	0,0050
k-GH-VNS1^G	***3 925,04***	3 925,04	***3 925,04***	***0,0000***

k-GH-VNS2^{G}	3 925,08	3 925,74	3 925,25	0,2756
k-GH-VNS3^{G}	3 925,08	3 926,32	3 925,29	0,5060
k-GH-VNS1-RND^{G}	***3 925,04***	3 925,04	***3 925,04***	***0,0000***
k-GH-VNS2-RND^{G}	3 925,07	3 925,12	3 925,09	0,0148
k-GH-VNS3-RND^{G}	3 925,05	3 925,08	3 925,07	0,0128
j-means^{G}	3 926,38	4 100,77	3 981,19	69,7025
j-means-GH-VNS1^{G}	no result	no result		
j-means-GH-VNS2^{G}	3 926,54	4 118,59	4 054,57	110,87
GA-FULL^{G}	3 925,07	3 925,10	3 925,08	0,0089
GA-MIX^{G}	***3 925,04***	3 925,08	3 925,05	0,0164
GA-ONE^{G}	***3 925,04***	3 925,07	3 925,05	0,0151

Clustering algorithms that perform better results for an objective function with a small number of clusters are not always the best as the number of clusters increases. However, the advantage of the family of greedy heuristic algorithms over the k-means algorithm, as well as j-means (considered one of the best) remains after the transition to the CUDA architecture. The application of the GPU shows an advantage in achieving speed compared to computing on a processor, and the advantage increases for large data sets and a large number of clusters by tens and hundreds of times [166, 167].

* * *

The greedy heuristics method [142] can be successfully applied in the formulation of an efficient combination of algorithms for solving the k-means problem.

New combined search algorithms with alternating randomized neighborhoods (k-GH-VNS) have more stable results, i.e., give a smaller minimum value and/or standard deviation of the objective function, a smaller spread of achieved values. Therefore, they have better indicators in comparison with known (classical) j-means and k-means algorithms. They also consistently show good results on all data sets, competing with genetic algorithms of the greedy heuristics method with the parallel implementation of new algorithms on the GPU for large automatic grouping problems adapted to the CUDA architecture. Parallel algorithms retain an important property of greedy heuristic algorithms such as high accuracy of the results obtained.

According to the results of computational experiments, the values of the objective function of new algorithms in some cases turn out to be significantly better and more stable than the results of the genetic algorithm. Nevertheless, new algorithms are inferior to genetic algorithms, but not significantly in some problems. It should be noted that new algorithms lose their advantage over genetic algorithms with a significant increase in computation time for medium size problems (up to approximately 10000 data vectors, up to 100 clusters). For large problems, the execution of only a few iterations of the genetic algorithm in an acceptable time (for example, a day) is not always possible even on modern computer technology, for example, massively parallel systems. At the same time, k-GH-VNS algorithms demonstrate good results.

Thus, the given chapter presents the solved problems of developing new search algorithms with alternating randomized neighborhoods for the k-means problem and the implementation of greedy heuristic algorithms for automatic grouping for massively parallel systems. It demonstrates that the parallel implementation of the local search algorithm, as well as individual steps of the greedy agglomerative heuristic procedure, makes it possible to construct an automatic grouping algorithm with a high acceleration factor. It reduces the calculation time by dozens of times without worsening the achieved value of the objective function.

Chapter 3. ALGORITHMS USING GREED AGGLOMERATIVE HEURISTIC PROCEDURES WITH ALTERNATED NEIGHBORHOODS FOR K-MEDOID PROBLEMS AND LIKELIHOOD FUNCTION MAXIMIZATION

The chapter is devoted to the development of combined algorithms of the method of greedy heuristics for problems of automatic grouping with increased requirements for accuracy and stability of the result using search algorithms with alternating randomized neighborhoods in relation to a wider range of problems: the problem of k-medoid and maximizing the likelihood function of mathematical expectation.

3.1 Combined search algorithms with alternating randomized neighborhoods for the k-medoid problem

One of the classic models of location theory is the p-medoid problem. The goal of the continuous placement problem [27] is to find the location of one or more points (depending on the specific problem setting - centers, centroids, medoids, etc.) in a continuous space (an infinite set of possible locations is considered desired points). There is also an intermediate class of problems that are actually discrete (the number of possible locations is finite), but at the same time operating with concepts that are characteristic of continuous problems. They include the p-medoid problem [168, 169] (also called the k-medoid problem or the discrete p-median problem [170] in the scientific studies). The main parameters of placement problems are the coordinates of objects and the distances between them [28, 171, 172].

The goal of the continuous p-median problem [171] is to find such k points (medoids, centers, centroids) that the sum of weighted distances from N known points (called requirement points, data vectors or consumers, depending on the setting and the subject area of the problem) to the nearest of k centers reached a minimum.

At present, a lot of algorithms have been proposed for solving the Weber problem for continuous location problems with Euclidean, Manhattan (rectangular), Cheby-shev metrics (all these metrics are special cases of metrics based on Minkowski's l_p-norms [173]). In particular, the well-known Weisfeld procedure [54] was generalized for metrics based on Minkowski norms.

In the case of a quadratic Euclidean metric for $L(X_j,A_i)=\sqrt{\sum_{k=1}^{d}(x_{j,k}-a_{i,k})^2}$ we have the k-means problem in the traditional representation. Here $X_j=(x_{j,1},\ldots,x_{j,k})$ $\forall j=\overline{1,p}$, $A_i=(a_{i,1},\ldots,a_{i,k})$ $\forall i=\overline{1,N}$. В случае квадратичной евклидовой метрики $L(X_j,A_i)=\sum_{k=1}^{d}(x_{j,k}-a_{i,k})^2$ at $w_i=1\forall i=\overline{1,N}$ we have the k-means problem. Here L is a distance function, N data vectors $A_1,\ldots,A_N$ in d-dimensional space $A_i=(a_{i,1},\ldots,a_{i,d})$, $A_i \in \mathbb{R}^d$.

The k-medoid problem (model) is different in that the centers of clusters, called medoids, are found exclusively among the known points A_i, i.e., it belongs to discrete optimization problems.

Local search methods for solving discrete optimization problems are the most natural and illustrative [139]. These ideas were also applied to solve location problems, salesman problems, networking, scheduling, etc. [174-177]. Simple local descent does not allow one to find the global optimum of the problem, but such methods are usually quite fast. The interest to them has not disappeared with the advent of new bionic algorithmic schemes and new theoretical results in the field of local search.

The classical local descent algorithm starts from the initial solution x_0 (in our case, from the initial set of medoids), which is chosen randomly or using some auxiliary algorithm. Until the local optimum is reached, at each step of the local descent, there is a transition from the current solution to a neighboring solution with a smaller value of the objective function.

The function of the neighborhood *O* at each step of the local descent sets the set of possible directions of the local search. Quite often this set contains several elements. There is a certain freedom when choosing the next solution, but the choice rule can have a significant impact on the result of the algorithm. In order to reduce the complexity of one step when choosing a neighborhood, it is desirable to have a set *O(X)* of as small a size as possible. Although, on the other hand, a wider neighborhood may lead to a better local optimum. One of the ways to solve this contradiction is the development of complex neighborhoods, the size of which can be varied in the course of a local search [43, 139].

The paper proposes a combined application of local search algorithms containing greedy agglomerative heuristic procedures, as well as the well-known PAM algorithm (Algorithm 1.2), applying a search scheme with alternating randomized neighborhoods [132, 178].

Algorithm 3.1 PAM-GH-VNS

```
1: Obtain a solution $S$ by running Algorithm 1.2 from a randomly generated initial solution.
2: $O=O_{start}$ (search neighborhood number).
3: $i=0, j=0$.
while $j < j_{max}$
    while $i < i_{max}$
        4: if the STOP conditions are not met, then get the solution S', by running Algorithm 1.2 from a random initial solution.
        repeat
        5: Depending on the value of $O$ (possible values are 1, 2, or 3), run Greedy Procedure Algorithm 1, 2, or 3, respectively, with initial solutions $S$ and S'. Thus, the neighborhood is determined by the method of including cluster centers from the second known solution and the parameter of the neighborhood is the second known solution.
                if a new solution is better than $S$, then
                write the new result to $S$, $i=0, j=0$.
            otherwise leave the loop.
        end of the cycle
        6: $i=i+1$.
    end of the cycle
    7: $i=0, j=j+1, O=O+1$, if $O>3$, then $O=1$.
end of the cycle
```

Further, the authors applied the new genetic algorithms GA-FULL and GA-ONE for comparative computational experiments. The description of the algorithms is given in [109, 142].

3.2. Results of computational experiments with new algorithms for the k-medoid problem

The tables below used the following abbreviations and abbreviations of the algorithms:

PAM is classic PAM algorithm;

PAM-GH-VNS1, PAM-GH-VNS2, PAM-GH-VNS3 are variations of the search algorithm with alternating randomized neighborhoods (Algorithm 2.8);

PAM-GH-VNS1-RND, PAM-GH-VNS2-RND, PAM-GH-VNS3-RND are ariations of the search algorithm with alternating randomized neighborhoods with a randomly determined initial solution;

GA-FULL is a genetic algorithm with a greedy heuristic with a real alphabet [109];

GA-ONE is a genetic algorithm that uses Algorithm 2.3 (Greedy Procedure 2) as the crossover procedure

As test data sets, the authors analyzed (Tables 3.1-3.3) the results of non-destructive test tests of prefabricated production batches of electrical and radio products, conducted in a specialized test center for completing the onboard equipment of spacecraft [178].

The DEXP OEM computing system (4-core Intel® Core™ i5-7400 CPU 3.00 GHz, 8 GB RAM) was used for the experiments.

For all data sets, 30 attempts were made to run each of the 9 algorithms. Only the best results achieved in each attempt were recorded, then from these results for each algorithm the objective function values were calculated such as minimum value (Min), mean value (Mean) and standard deviation.

The results of the computational experiments are presented in Tables 3.1-3.3. The best objective function values (minimum value, mean value, and standard deviation) are in bold italics.

Algorithms can be compared by the mean value of the objective function or by the median. In this study, the authors use the mean value of the objective function, because if some algorithm randomly produces the best result of the objective function once (minimum value, mean value or standard deviation), and in other experiments it is worse, then such an algorithm of course, will not be the best.

A graphical comparison of new and known algorithms for each data set (Tables 3.1-3.3) is shown in the convergence graphs of algorithms built on the average value of the objective function (Figures 3.1-3.3). On the abscissa, it is time, on the ordinate, it is the achieved mean value of the objective function.

Table 3.1

Results of computational experiments on the 3OT122A data set (767 data vectors, each dimension 13) 10 clusters, 60 seconds, 30 attempts, Manhattan distance

Algorithm	Objective function value		
	Min (record)	mean	Root mean square deviation
PAM	1 654,39	1 677,35	12,2445
PAM-GH-VNS1	***1 554,26***	***1 566,70***	7,4928
PAM-GH-VNS2	1 558,02	***1 566,06***	***5,0686***
PAM-GH-VNS3	***1 555,09***	***1 563,99***	***3,9161***
GA-FULL	1 599,23	1 637,58	25,5365
GA-ONE	1 589,87	1 614,78	13,5342

Table 3.2

Results of computational experiments on data set 5514BC1T2-9A5 (91 data vectors, each dimension 173) 10 clusters, 60 seconds, 30 attempts, Manhattan distance

Algorithm	Objective function value		
	Min (record)	Mean	Root mean square deviation
PAM	7 623,81	7 629,74	8,4124
PAM-GH-VNS1	***7 604,49***	***7 604,49***	***0,0000***
PAM-GH-VNS2	***7 604,49***	***7 604,49***	***0,0000***
PAM-GH-VNS3	***7 604,49***	***7 604,49***	***0,0000***
GA-FULL	***7 604,49***	***7 604,49***	***0,0000***
GA-ONE	***7 604,49***	7 606,43	6,1597

Table 3.3

Results of computational experiments on data set 1526TL1 (1234 data vectors, each dimension 157) 10 clusters, 60 seconds, 30 attempts, Manhattan distance

Algorithm	Objective function value		
	Min (record)	Mean	Root mean square deviation
PAM	50 184,01	50 883,73	472,4409
PAM-GH-VNS1	***45 440,37***	***45 553,02***	***95,8004***
PAM-GH-VNS2	45 453,68	***45 657,68***	153,3286
PAM-GH-VNS3	***45 444,42***	***45 637,87***	177,5864
GA-FULL	46 660,86	48 391,17	845,0838
GA-ONE	47 081,34	48 125,99	766,5659

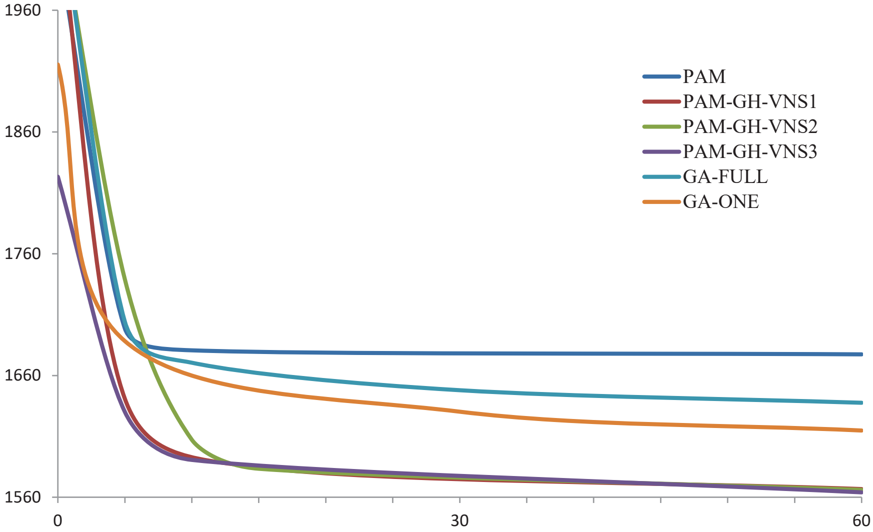


Fig. 3.1. Comparison of new and known algorithms for data set 3OT122A (10 clusters, 60 seconds): along the abscissa axis - time in seconds; along the ordinate axis - the achieved average value of the objective function

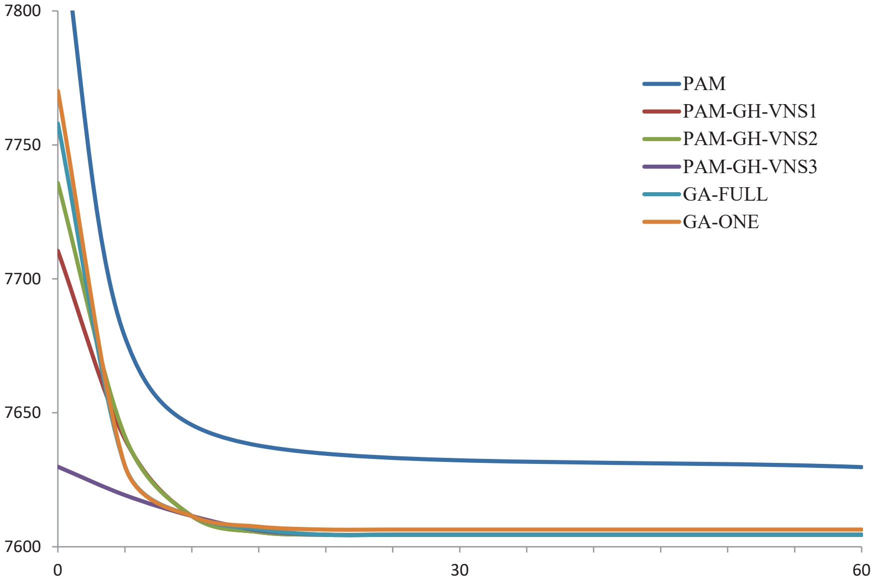


Fig. 3.2. Comparison of new and known algorithms for data set 5514BC1T2-9A5 (10 clusters, 60 seconds): along the abscissa axis - time in seconds, along the ordinate axis - the achieved average value of the objective function

The results of earlier computational experiments on data sets of electrical radio products with various modifications of the genetic algorithm were used for a more complete comparison of the obtained results of computational experiments [142]. Prefabricated batches of 1526TL1 electrical and radio products were used for calculations.

Table 3.4 uses the following abbreviations [142]. They are GA - genetic algorithm, ZhE - greedy heuristic, GAHE - genetic algorithm with greedy heuristic with real alphabet, LP - local search, GA FP - genetic algorithm with recombination of subsets of a fixed length [37] , IBC – Information Bottleneck Clustering, JL – multistart greedy heuristic with local search enabled, k-mean. multistart – multistart of the ALA procedure.

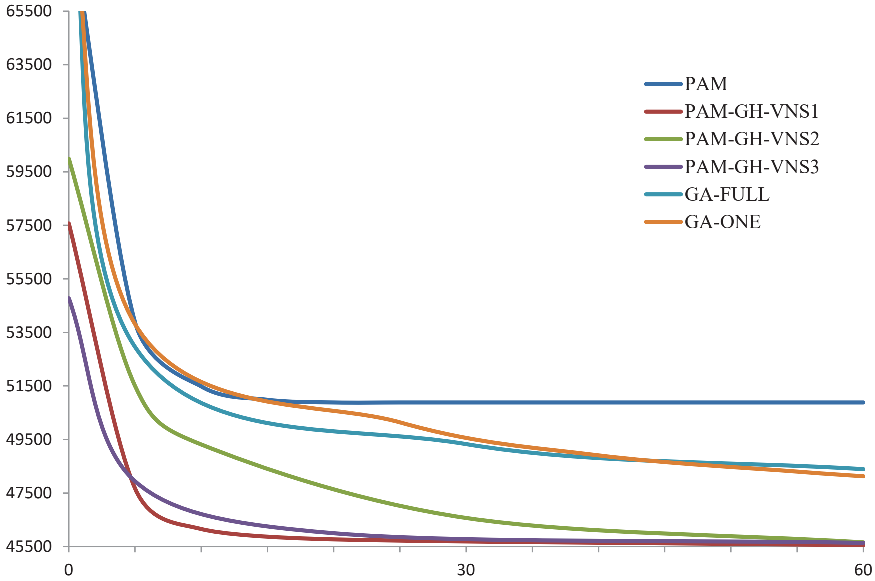


Fig. 3.3. Comparison of new and known algorithms for dataset 1526TL1 (10 clusters, 60 seconds):
along the abscissa axis - time in seconds; along the ordinate axis - achieved average value of the objective function

The authors used publicly available and well-known data sets from the UCI repositories [155] and clustering basic bench-mark for the final verification and conclusions on the results of computational experiments with production batches of electrical and radio products for spacecraft and for the possibility of applying our new algorithms [156].

Moreover, calculations were made with a different number of clusters and different distances (Tables 3.5-3.7). Figures 3.4-3.6 present graphical implementation of the convergence of algorithms built on the average value of the objective function.

Table 3.4

Results of computational experiments on data set 1526TL1 (1234 data vectors, each dimension 157) 10 clusters, 60 seconds, 30 attempts, squared Euclidean distance

Algorithm	Objective function value		
	Min (record)	Mean	Root mean square deviation
PAM	64 232,02	66 520,18	991,9938
PAM-GH-VNS1	55 373,00	55 906,02	416,4050
PAM-GH-VNS2	***55 361,75***	55 858,35	359,4161
PAM-GH-VNS3	55 383,81	55 755,00	353,9469
GA-FULL	58 789,34	60 629,52	1 187,0953
GA-ONE	58 300,15	60 165,43	1 388,6209
GAGH+LP	***55 361,75***	55 364,10	6,2204
GAGH real., σe=0.25	***55 361,75***	***55 361,75***	***7,86E-12***
GAGH real partially, σe=0.25	***55 361,75***	***55 361,75***	***7,86E-12***
GA FS	***55 361,75***	55 452,68	240,5632
GA classical	***55 361,75***	55 364,10	6,2204
IBC, σe=0.25	no result		
Determ. GH, σe=0.25	57 131,00	57 131,00	***0,0000***
Determ. GH, σe=0.001	55 998,22	55 998,22	***0,0000***
IBC, σe=0.001	no result		
GH adaptive, σe=0.25	***55 361,75***	***55 361,75***	***7,86E-12***
GH adaptive, σe=0.001	***55 361,75***	55 381,31	33,3953
GH, σe=0.25, β=0.5	***55 361,75***	***55 361,75***	***7,86E-12***
GH, σe=0.25, β=1	***55 361,75***	***55 361,75***	***7,86E-12***
GH, σe=0.25, β=3	***55 361,75***	55 371,53	25,8679
GH, σe=0.001, β=0.5	***55 361,75***	55 366,45	8,0305
GH, σe=0.001, β=1	***55 361,75***	***55 361,75***	***7,86E-12***
GH, σe=0.001, β=3	***55 361,75***	55 371,53	25,8679
GL, σe=0.25, β=0.5	***55 361,75***	55 604,47	294,1579
GL, σe=0.25, β=1	***55 361,75***	55 455,03	239,6050
GL, σe=0.25, β=3	***55 361,75***	55 907,30	240,5632
GL, σe=0.001, β=0.5	***55 361,75***	55 548,22	241,5122
GL, σe=0.001, β=1	***55 361,75***	55 634,52	340,2077
GL, σe=0.001, β=3	***55 361,75***	55 907,30	240,5632
k-mean multistart	***55 361,75***	55 364,10	6,220381

Table 3.5

Results of computational experiments on the ionosphere dataset (351 data vectors, each dimension 35) 10 clusters, 60 seconds, 30 attempts, Manhattan distance

Algorithm	Objective function value		
	Min (record)	Mean	Root mean square deviation
PAM	2 688,57	2 704,17	12,3308
PAM-GH-VNS1	***2 607,21***	***2 607,25***	***0,1497***
PAM-GH-VNS2	***2 607,21***	***2 607,43***	***0,4303***
PAM-GH-VNS3	***2 607,21***	***2 607,34***	***0,4159***
GA-FULL	***2 608,22***	2 624,97	9,5896
GA-ONE	***2 608,69***	2 625,18	10,7757

Table 3.6

Results of computational experiments on the Mopsi-Joensuu dataset (6015 data vectors, each dimension 2) 20 clusters, 60 seconds, 30 attempts, Euclidean distance

Algorithm	Objective function value		
	Min (record)	Mean	Root mean square deviation
PAM	319,84	343,44	***15,3004***
PAM-GH-VNS1	278,63	390,43	82,6086
PAM-GH-VNS2	333,26	471,15	100,2594
PAM-GH-VNS3	***273,91***	***334,98***	51,2440
PAM-GH-VNS1-RND	301,91	428,14	129,2156
PAM-GH-VNS2-RND	384,62	475,92	53,0972
PAM-GH-VNS3-RND	***265,96***	***325,49***	42,1440
GA-FULL	315,57	383,41	60,1489
GA-ONE	343,21	433,01	66,0360

Table 3.7

Results of computational experiments on the Chess dataset (3196 data vectors, each dimension 37) 50 clusters, 60 sec, 30 tries, squared Euclidean distance

Algorithm	Objective function value		
	Min (record)	Mean	Root mean square deviation
PAM	10 763,0	10 822,4	***47,1268***
PAM-GH-VNS1	10 357,0	***10 530,9***	122,9620
PAM-GH-VNS2	10 803,0	11 107,1	174,1184
PAM-GH-VNS3	10 429,0	10 594,6	114,7192
PAM-GH-VNS1-RND	10 400,0	10 659,0	161,2982
PAM-GH-VNS2-RND	10 891,0	11 097,0	187,9113
PAM-GH-VNS3-RND	***10 310,0***	10 623,3	214,1288
GA-FULL	***10 252,0***	***10 381,3***	***72,9110***
GA-ONE	10 944,0	11 098,0	112,0813

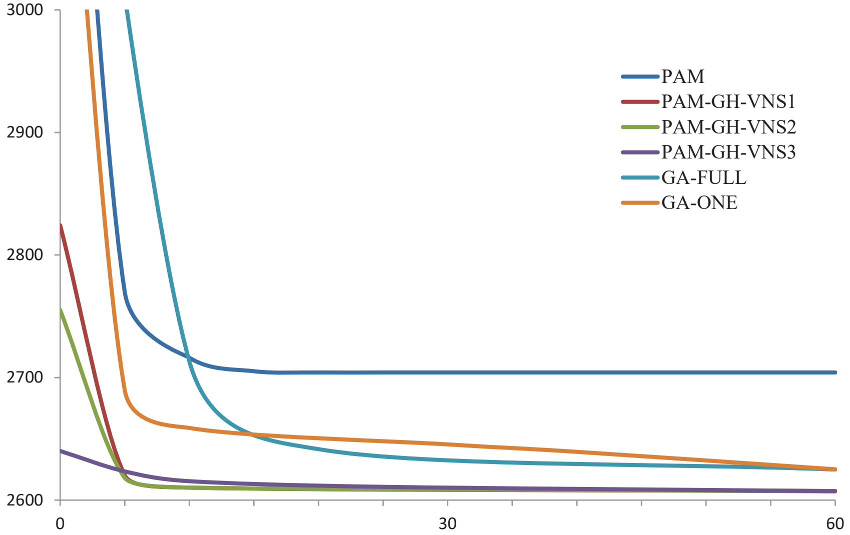


Fig. 3.4. Comparison of new and known algorithms for the ionosphere dataset (10 clusters, 60 seconds, Manhattan distance): along the abscissa axis - time in seconds; along the ordinate axis - the achieved average value of the objective function

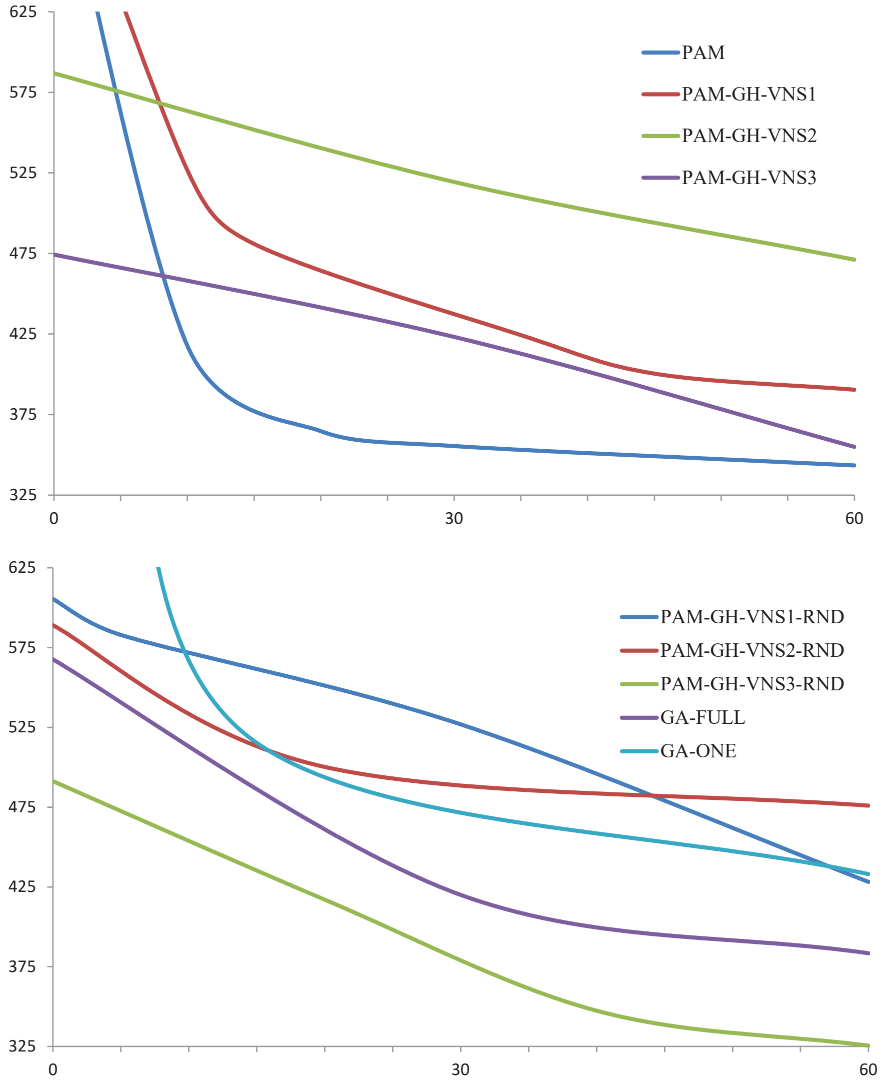


Fig. 3.5. Comparison of new and known algorithms for the Mopsi-Joensuu dataset (20 clusters, 60 seconds, Euclidean distance): along the abscissa axis - time in seconds; along the ordinate axis - the achieved average value of the objective function

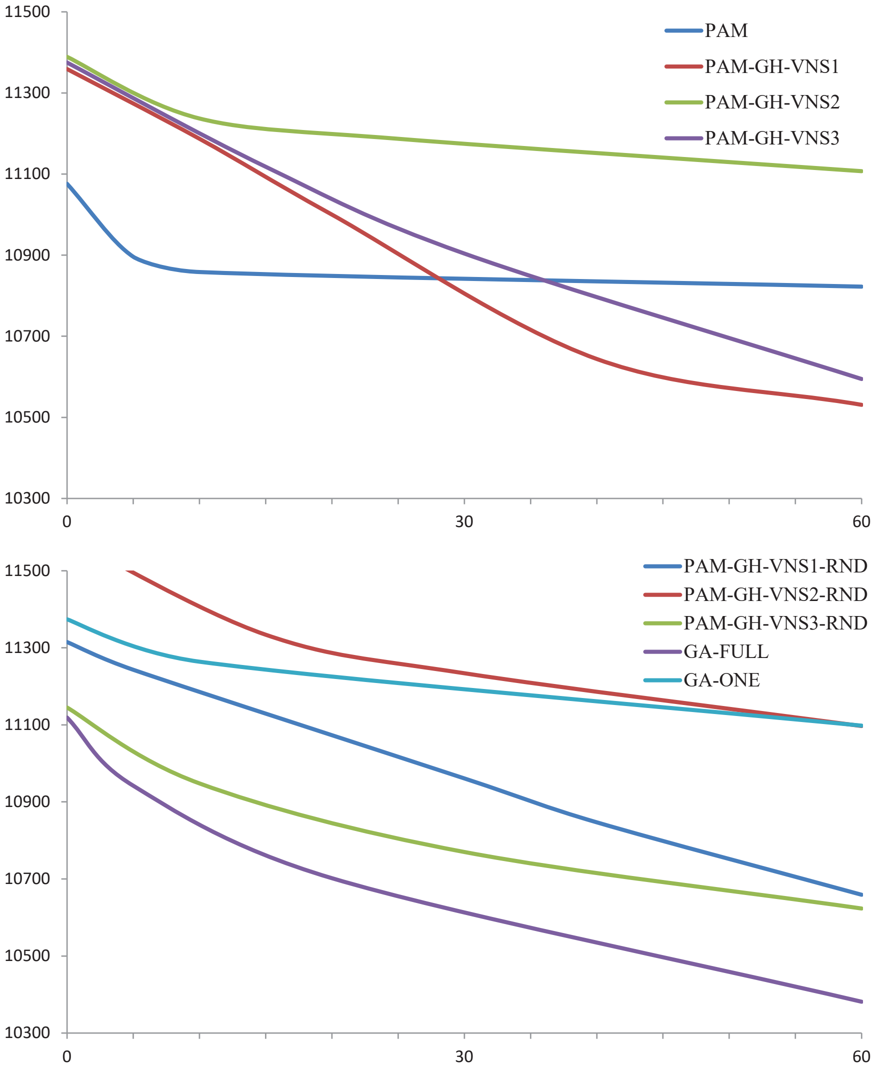


Fig. 3.6. Comparison of new and known algorithms for the Chess dataset (50 clusters, 60 seconds, squared Euclidean distance): along the abscissa axis - time in seconds; along the ordinate axis - the achieved average value of the objective function

The results of computational experiments have shown that new combined search algorithms with alternating randomized neighborhoods (PAM-GH-VNS) with a small number of clusters have more stable results (they give a lower mean value and/or standard deviation of the achieved value of the objective function, smaller times - reset of achieved values) and, therefore, better performance in comparison with known algorithms (on ionosphere datasets and electrical radio products). When the number of on Boolean data is more than 20 clusters both new and genetic algorithms of the greedy heuristics method having an advantage in achieving the best average achieved values of the objective function with multiple runs, do not demonstrate advantages in terms of the stability of the obtained value objective function (PAM-algorithm shows a more stable, but consistently poor result compared to some of new algorithms).

3.3. Combined classification EM algorithm

Currently, there exist a great number of data clustering methods [180]. The EM-algorithm (Expectation Maximization, i.e., maximization of mathematical expectation) is one of the popular ones. It is used in case of analysis of incomplete data [107, 110, 181]:

- for some reason, some statistics are missing;
- likelihood function has a form that allows for serious simplifications with the introduction of additional "hidden" quantities, but does not allow for "convenient" research methods.

It is reduced to the task of separating a mixture of probability distributions in solving the clustering problem by the EM-algorithm. The general statement of the problem of separation of a mixture of distributions is as follows.

Let the distribution density on the set *X* have the form of a mixture of *k* distributions (assume that the distributions are Gaussian):

$$\rho(x) = \sum_{j=1}^{k} \alpha_j \rho_j(x), \ \sum_{j=1}^{k} \alpha_j = 1, \ \alpha_j \geq 0,$$

where $\rho_j(x)$ is the likelihood function of the *j*-th component of the mixture, α_j is its a priori probability (“weight” in the composition of the mixture).

The main advantage of the EM-algorithm is its ease of the performance. Moreover, it can optimize not only model parameters, but also make assumptions about missing data values.

This makes EM an excellent method for clustering and creating parameterized models one can assume what the cluster contains and where the new data should be attributed knowing the clusters and the parameters of the model.

Although the EM-algorithm has its disadvantages:

1. The performance of the algorithm decreases as the number of iterations increases.
2. EM does not always find the optimal parameters and can get stuck in the local optimum without finding the global one.

The EM-algorithm is the so-called "greedy" algorithm. Its essence is to make locally optimal decisions at each stage. A local maximum can be very different from the global one. The CEM algorithm (Classification EM) implements a randomized, but purposeful “shaking” of the sample at each iteration for this. This helps to “knock out” the optimization process from local maxima.

The deterministic rule works in the CEM algorithm. According to it, an object is assigned to one cluster, the number of which matches the number of the largest of the numbers. In general, CEM works relatively fast compared to EM. As a rule, CEM finds an extreme close to the global one.

In order to compare the results of EM and CEM algorithms, a study was conducted to check the significance of information features that supposedly have an exponential distribution for the problem of selecting homogeneous batches of electrical radio products (detailed in Chapter 4).

The initial data for analysis in solving the problem are the results of test effects on electrical and radio products to control the current-voltage characteristics of the input and output circuits of microcircuits.

The number of errors (an error is an incorrectly defined batch) when running the EM and CEM algorithms is presented in Table 3.8. The denominator of the fraction is the volume of the combined party.

According to Table 3.8, CEM works on average 10% better than EM. It has been experimentally established that the main EM algorithm is more unstable in terms of initial data [107, 109, 182].

Table 3.8

Results of EM and CEM algorithms (average errors over 10 runs)

Algorithm	Chips 1526TL1 (4 features, 1 exponential) 3 parties	Chips 1526IE10 (6 features, 2 exponential) 5 parties	Diodes 3OT122A (4 features, 1 exponential) 3 parties
EM	152/626	217/870	68/279
CEM	96/626	130/870	40/279

According to the table CEM works on average 10% better than EM. It has been experimentally established that the main EM algorithm is more unstable in terms of initial data [107, 109, 182].

Algorithm 3.2 CEM algorithm (classification EM algorithm)

A sample (array) of N vectors of d-dimensional data $X_i = \left(x_{i,1}, \dots, x_{i,d}\right)^T, i = \overline{1,N}$, estimated number of distributions in a mixture of k.

Step 1 (initialization). Select some initial values of the distribution parameters. As a rule, the values of randomly selected data vectors are chosen as the expectation vectors μ, and the values of the variances (or covariance matrices) are set the same for all distributions and calculated for the entire sample, or unit matrices are taken as covariance matrices (similarly, for exponential distributions or Laplace distributions, the parameter α is calculated over the entire sample $X_1,\dots,X_N$).

Set the values of prior probabilities of each of the distributions equal for all distributions $w_j = 1/k, j = \overline{1,k}$.

Step 2 (E-step - classification/clustering).

With fuzzy clustering, for each distribution j and for each data vector i, the posterior probability that the i-th data vector belongs to the j-th distribution is calculated: $g_{i,j} = \frac{f(x_i|j)w_j}{\sum_{l=1}^{k}(f(x_i|l)w_l)} \forall i = \overline{1,N}, j = \overline{1,k}$.

Here $f(x_i|j)$ is the density of the j-th distribution at the point x_i.

When the CEM-algorithm is fulfilled for each data vector, all values of gi,j for all distributions are set to 0, except for one distribution j', for which $\frac{f(x_i|j')w_{j'}}{\sum_{l=1}^{k}(f(x_i|l)w_l)}$ *has a maximum value. This distribution is set to*

$g_{i,j'} = 1$. *Consider that the data vectors for which* $g_{i,j'} = 1$, *form the j'th cluster.*

Step 3 (M-step, i.e., modification of distribution parameters).

3.1. Recalculate the values of aprior probabilities:

$$w_j = \frac{\sum_{i=1}^{N} g_{i,j}}{N} \forall j = \overline{1,k}.$$

3.2. Recalculate the estimates of the parameters of each of the distributions, taking into account the posterior probability that a specific i-th data vector is included in the j-th cluster with a probability $g_{i,j}$. For example, the vector of average values $\mu_j = (\mu_{j,1}, \dots, \mu_{j,d})$ for each cluster is calculated by the formula

$$\mu_{j,l} = \frac{1}{\sum_{q=1}^{N} g_{q,j}} \sum_{i=1}^{N} x_{i,l} g_{i,j} = \frac{1}{N w_j} \sum_{i=1}^{N} x_{i,l} g_{i,j} \quad \forall j = \overline{1,d}, l = \overline{1,k}.$$

Similarly, estimates of standard deviations are calculated as follows:

$$\sigma_{j,l}{}^2 = \frac{1}{\sum_{q=1}^{N} g_{q,j}} \sum_{i=1}^{N} (x_{i,l} - \mu_{j,l})^2 g_{i,j} = \frac{1}{N w_j} \sum_{i=1}^{N} (x_{i,l} - \mu_{j,l})^2 g_{i,j} \quad \forall j = \overline{1,d}, l = \overline{1,k}.$$

Here $\sigma_{j,l}$ is the standard deviation for the *l*-th dimension in the *j*-th distribution (cluster).

When fulfilling the CEM algorithm, the mean vector and standard deviations are calculated for each cluster separately. It is easy to prove that the two formulas above are also suitable for the CEM case, but the calculation for each cluster separately is faster and more accurate in practice.

The application of the standard multivariate distribution with a full covariance matrix is as follows:

$$\Sigma(j) = \begin{pmatrix} \sigma(j)_1^2 = \sigma_{1,1} & \sigma(j)_{1,2} & \dots & \sigma(j)_{1,d} \\ \sigma(j)_{2,1} & \sigma(j)_2^2 = \sigma(j)_{2,2} & \dots & \sigma(j)_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(j)_{d,1} & \sigma(j)_{d,2} & \dots & \sigma(j)_d^2 = \sigma(j)_{d,d} \end{pmatrix}$$

Its elements are also calculated taking into account a posteriori probability:

$$\sigma(j)_{p,q} = \sigma(j)_{q,p} = \frac{1}{N w_j} \sum_{i=1}^{N} (x_{i,p} - \mu_{j,p}) (x_{i,q} - \mu_{j,q}) g_{i,j}.$$

4. Calculate the value of the objective function, i.e., the logarithmic likelihood function:

$$Q(w_1, \dots, w_1, (parameters\ of\ all\ distributions) = \sum_{i=1}^{N} ln(\sum_{j=1}^{k} w_j f(x_i|j))$$

5. Check stop conditions and go to Step 2.

The application of various probabilistic models in the EM-algorithm for the problem of dividing batches of industrial products into homogeneous batches was studied in [109]. It presents that in the case of multivariate data, the model with multivariate uncorrelated Gaussian measurements is the most adequate (gives the least number of errors when tested on data sets with pre-labeled data) in comparison with multivariate Gaussian distributions with a full covariance matrix and in comparison, with spherical Gaussian distributions.

A multivariate Gaussian distribution with independent (uncorrelated) features (dimensions) differs from a multivariate Gaussian distribution only in that it does not require operation with matrices.

Given: N vectors of d-dimensional data $X_i = \left(x_{i,1}, \dots, x_{i,d}\right)^T \quad, i = \overline{1, N}$.

There is a vector $\mu \in \mathbb{R}^d$ and a non-negative definite symmetric covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 = \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,d} \\ \sigma_{2,1} & \sigma_2^2 = \sigma_{2,2} & \cdots & \sigma_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d,1} & \sigma_{d,2} & \cdots & \sigma_d^2 = \sigma_{d,d} \end{pmatrix} \mathrm{p}$$

dimensions $d \times d$ such that the probability density of the vector X are as follows:

$$f(X) = \frac{\alpha}{\sqrt{(2\pi)^d |\Sigma|}} exp(-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)),$$

where | Σ | is the determinant of the matrix Σ, and Σ^{-1} is the matrix inverse to Σ. The components of the vector μ are calculated separately for each dimension: $\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{i,j}$.

Here $x_{i,j}$ is the j-th component (j-th dimension) of the i-th data vector.

The covariance matrix is diagonal (therefore, it can be replaced by a vector). It consists of the variances for each dimension:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_d^2 \end{pmatrix}.$$

Variances are calculated independently: $\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N} \quad (x_{i,j} - \mu_j)^2$.

The distribution density is calculated as the product of the densities for each dimension:

$$f(X) = \prod_{j=1}^{d} f_j(x_j) = \prod_{j=1}^{d} \frac{1}{\sigma_j\sqrt{(2\pi)}} exp(-\frac{1}{2}(x_j - \mu_j)^2/\sigma_j^2),$$

where is the jth component of the vector X.

The CEM-algorithm, as a modification of the EM-algorithm, can be quite successfully used as a local search method [107, 109, 112, 183, 184]. Solutions formed from elements of different solutions that are local optima are more likely to be closer to the global optimum in comparison to randomly selected solutions [110]. Therefore, in this case, it was also proposed to apply the VNS-algorithm as an extended local search [150, 183, 184].

Thus, an improved algorithm based on the classification EM-algorithm (Algorithm 2.9) applying a search with alternating randomized neighborhoods will be as follows [183, 184]:

Algorithm 3.3 CEM-GH-VNS

```
1: Obtain a solution S by running the CEM-algorithm from a randomly
   generated initial solution.
2: O=O_start (number of the search neighborhood).
3: i=0, j=0.
while j < j_max
    while i < i_max
        4: if the STOP conditions are not fulfilled, then get the solution
           S' by running the CEM-algorithm from a random initial solution.
        repeat
```

```
          5: According to the value of the variable $O$ (possible values
             are 1, 2 or 3), run the Greedy Procedure Algorithm 1, 2
             or 3 respectively with initial solutions $S$ and $S'$. Thus, the
             neighborhood is determined by the method of including
             cluster centers from the second known solution and the
             parameter of the neighborhood is the second known solution.
             if a new solution is better than S, then
             write the new result to $S$, $i=0$, $j=0$.
          otherwise leave the loop.
      end of the cycle
      6: $i=i+1$.
    end of the cycle
    7: $i=0$, $j=j+1$, $O=O+1$, if $O>3$, then $O=1$.
  end of the cycle
```

As test data sets, the authors applied (Table 3.9) the results of non-destructive test tests of prefabricated production batches of electrical and radio products, conducted in a specialized test center for completing the onboard equipment of spacecraft.

The DEXP OEM computing system (4-core Intel® Core™ i5-7400 CPU 3.00 GHz, 8 GB RAM) was used for the experiments.

For all data sets, 30 attempts were made to run each of the algorithms. Only the best results achieved in each attempt were recorded, then from these results for each algorithm the objective function values were calculated: the minimum value (Min), the average value (Mean) and the standard deviation. The best objective function's values (minimum value, mean value, and standard deviation) are in bold italics. Figures 3.7-3.9. demonstrate graphical implementation of the convergence of algorithms built on the average value of the objective function.

Table 3.9

Results of computational experiments on test data sets of a batch of industrial products (10 clusters, 2 minutes, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
3OT122A (767 data vectors, each dimension is 13)				
CEM	120 947,6	146 428,5	135 777,6	***7 985,6992***
CEM-GH-VNS1	121 256,5	152 729,1	***143 956,0***	8 708,6293
CEM-GH-VNS2	123 664,4	158 759,2	***143 028,5***	10 294,3992
CEM-GH-VNS3	***128 282,2***	155 761,9	***143 506,9***	10 058,8266
1526TL1 (1234 data vectors, each dimension is 157)				
CEM	354 007,3	416 538,4	384 883,4	***20 792,8068***
CEM-GH-VNS1	376 137,1	477 124,5	438 109,4	29 964,0641
CEM-GH-VNS2	345 072,6	487 498,3	444 378,1	43 575,3282
CEM-GH-VNS3	***379 352,3***	516 777,8	***456 271,4***	38 323,0246
5514BC1T2-9A5 (91 data vectors, each dimension is 173)				
CEM	4 504,1	7 284,2	5 776,4	987,9598
CEM-GH-VNS1	3 977,6	9 620,5	***6 981,3***	1 990,3690
CEM-GH-VNS2	***4 528,9***	13 545,5	6 342,4	2 632,7929
CEM-GH-VNS3	4 415,6	7 112,9	5 966,3	***904,9495***

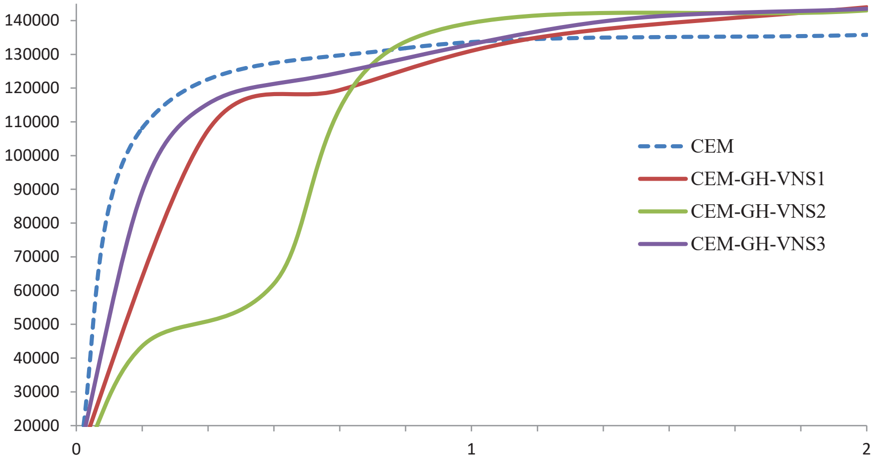


Fig. 3.7. Comparison of new and known algorithms for data set 3OT122A (10 clusters, 2 minutes):
along the abscissa axis - time in minutes; along the ordinate axis - the achieved average value of the objective function

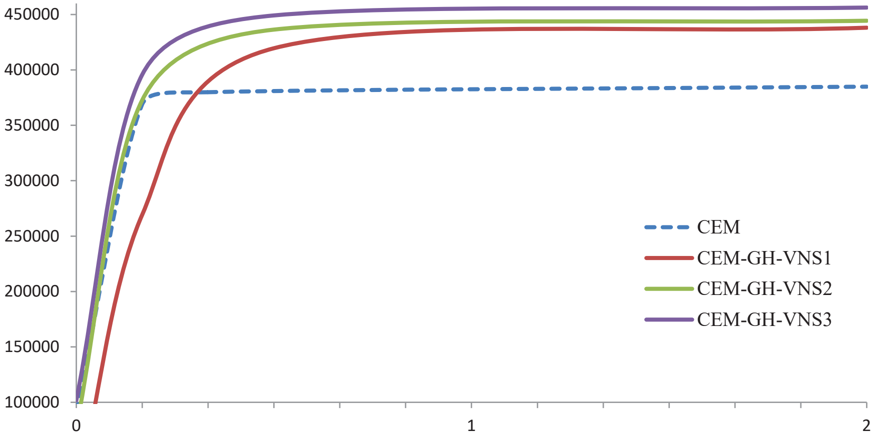


Fig. 3.8. Comparison of new and known algorithms for data set 1526TL1 (10 clusters, 2 minutes):
along the abscissa axis - time in minutes; along the ordinate axis - the achieved average value of the objective function

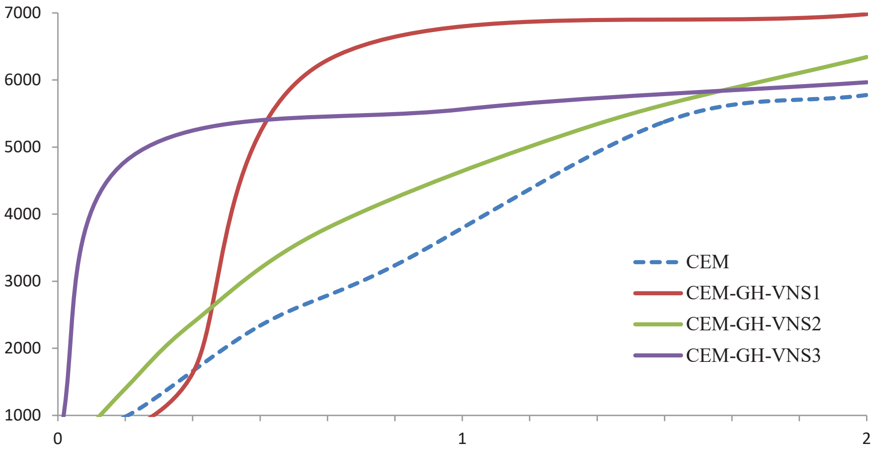


Fig. 3.9. Comparison of new and known algorithms for data set 5514BC1T2-9A5 (10 clusters, 2 minutes):
along the abscissa axis - time in minutes; along the ordinate axis - the achieved average value of the objective function

As computational experiments show [148, 182-185], the stability of the results with multiple runs of the CEM-algorithm is higher (the standard deviation of the objective function is smaller) than that of new algorithms, at the same time, the result in many cases is far from the true optimum of the likelihood function. The available opportunities for improving the results are evidenced by the fact that when conducting repeated computational experiments, the results of the best attempts to run the CEM-algorithm sometimes differ by tens of percent in terms of the value of the objective likelihood function from the averaged values of the entire set of attempts. Therefore, new search algorithms with alternating randomized neighborhoods (CEM-GH-VNS) have an advantage over the classical CEM-algorithm in terms of the average value of the objective function achieved during multiple runs.

3.4. Approach to the development of clustering algorithms based on parametric optimization models

Thus, the authors considered combinations of greedy algorithms with alternating neighborhoods for k-means, k-medoid problems, and well-known j-means and CEM-algorithms.

Figure 3.10 presents a flowchart of a new approach to the development of automatic grouping algorithms based on parametric optimization models, with the combined use of alternating randomized neighborhood search algorithms and greedy agglomerative heuristics.

A general scheme of the proposed new approach to the development of automatic grouping algorithms based on parametric optimization models with the combined use of search algorithms with alternating randomized neighborhoods and greedy agglomerative heuristic procedures can be described as follows:

Algorithm 3.4 GH-VNS (Greedy Heuristics in the Variable Neighborhood Search)

```
1: Obtain a solution $S$ by running a two-step local search algorithm
   from a randomly generated initial solution.
2: $O=O_{start}$ (search neighborhood number).
3: $i=0, j=0$ (number of unsuccessful iterations in a particular neighborhood
   and in general according to the algorithm).
while $j < j_{\max}$
```

```
    while $i < i_{\max}$
        4: if the STOP conditions are not met (exceeding the time limit),
           then get the solution $S'$ by running a two-step local search
           algorithm from a random initial solution.
        repeat
            5: Depending on the value of $O$ (possible values are 1, 2 or
               3), run Greedy Procedure Algorithm 1 or 2 or 3 respectively
               with initial solutions $S$ and $S'$. Thus, the neighborhood
               is determined by the method of including cluster
               centers from the second known solution and the parameter
               of the neighborhood is the second known solution.
               if the new solution is better than $S$, then
               write the new result to $S$, $i$=0, $j$=0.
            otherwise leave the loop.
        end of the cycle
        6: $i$=$i$+1.
    end of the cycle
    7: $i$=0, $j$=$j$+1, $O$=$O$+1, if $O$>3, then $O$=1.
end of the cycle
```

Depending on the value of O_{start} the algorithms in this study are designated GH-VNS1, GH-VNS2, GH-VNS3 (for the k-means problem, respectively, k-GH-VNS1, k-GH-VNS2, k-GH-VNS3; for solving the p-medoid problem: PAM-GH-VNS1, PAM-GH-VNS2, PAM-GH-VNS3; for solving problems using the CEM algorithm: CEM-GH-VNS1, CEM-GH-VNS2, CEM-GH -VNS3).

* * *

The results of computational experiments have shown that new algorithms of the greedy heuristic method for automatic grouping problems with increased requirements for the accuracy of the result (by the value of the objective function), using search algorithms with alternating randomized neighborhoods (GH-VNS) have more stable (lower standard deviation of the objective function) and more accurate (lower average value of the objective function) results, and therefore better performance compared to classical algorithms (k-means, j-means, PAM and CEM).

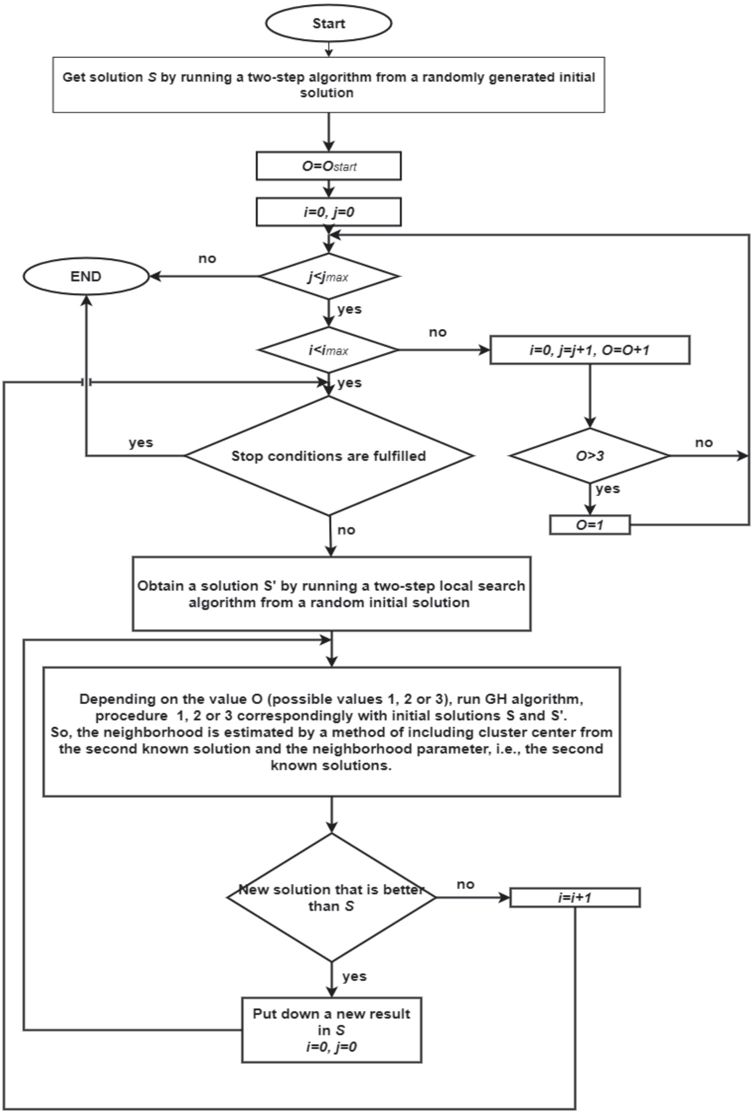

Start
Get solution S by running a two-step algorithm from a randomly generated initial solution
O=Ostart
i=0, j=0
j<jmax
no
END
yes
i<imax
no
i=0, j=j+1, O=O+1
yes
yes
Stop conditions are fulfilled
O>3
no
yes
O=1
no
Obtain a solution S' by running a two-step local search algorithm from a random initial solution
Depending on the value O (possible values 1, 2 or 3), run GH algorithm, procedure 1, 2 or 3 correspondingly with initial solutions S and S'. So, the neighborhood is estimated by a method of including cluster center from the second known solution and the neighborhood parameter, i.e., the second known solutions.
New solution that is better than S
no
i=i+1
yes
Put down a new result in S
i=0, j=0

Fig. 3.10. General scheme of the approach to the development of automatic grouping algorithms based on parametric optimization models, with the combined use of search algorithms with alternating randomized neighborhoods and greedy agglomerative heuristic procedures

At the same time, with an increase in the number of clusters and sample size, the comparative efficiency of the new approach based on parametric optimization models with the combined use of search algorithms with alternating randomized neighborhoods and greedy agglomerative heuristic procedures increases, and for large sets, these new algorithms have an advantage for a fixed running time of the algorithm.

However, it should be noted that with a significant increase in the calculation time, the known genetic algorithms of the greedy heuristic method show, albeit slightly, better results in comparison with the proposed new algorithms. Nevertheless, one can talk about the competitiveness of new algorithms both in comparison with the classical algorithms of k-means, PAM and j-means, and with genetic algorithms, including algorithms of the greedy heuristic method, as well as with deterministic algorithms.

Figure 3.11 shows a flowchart of the greedy heuristic method with the addition of new components developed as part of this study. The vertical order of the components reflects the nesting of the algorithms.

A new approach to the development of automatic grouping algorithms based on parametric optimization models, with the combined use of search algorithms with alternating randomized neighborhoods and greedy agglomerative heuristic procedures, was used in the activities of JSC Testing Technical Center - NPO PM.

Currently, cluster analysis tends to use collective methods [186]. Cluster analysis algorithms are not universal. Each algorithm has its own specific area of application. In the case, the area under consideration contains various types of data sets, it is necessary to apply not one specific algorithm, but a set of different algorithms to select clusters. The ensemble (collective) approach makes it possible to reduce the dependence of the final solution on the chosen parameters of the initial algorithms and obtain a more stable solution [187-190]. Chapter 4 will consider ensembles of automatic grouping algorithms.

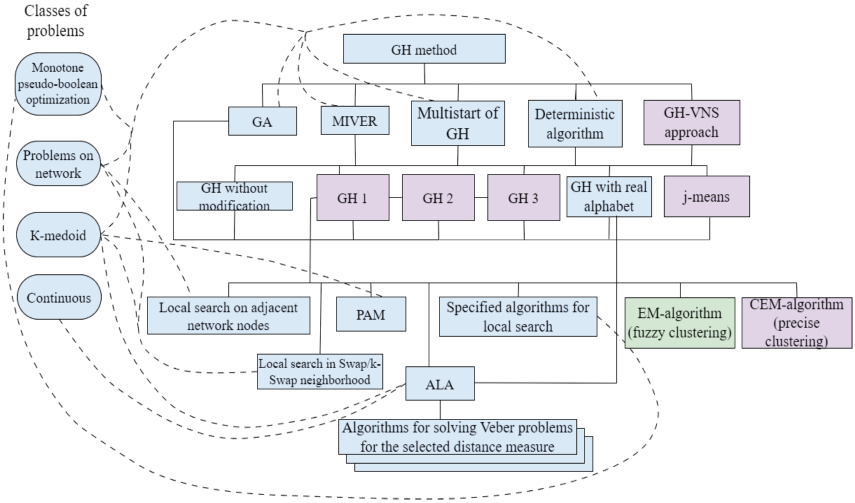

Classes of problems
Monotone pseudo-boolean optimization
Problems on network
K-medoid
Continuous
GH method
GA
MIVER
Multistart of GH
Deterministic algorithm
GH-VNS approach
GH without modification
GH 1
GH 2
GH 3
GH with real alphabet
j-means
Local search on adjacent network nodes
PAM
Specified algorithms for local search
EM-algorithm (fuzzy clustering)
CEM-algorithm (precise clustering)
Local search in Swap/k-Swap neighborhood
ALA
Algorithms for solving Veber problems for the selected distance measure

Fig. 3.11. Components of the greedy heuristic method, their mutual compatibility (solid lines) and applicability to problem classes (italic lines). New components are highlighted in purple

Chapter 4. APPLICATION OF GREEDY AGGLOMERATIVE HEURISTIC PROCEDURES IN PROBLEMS OF AUTOMATIC GROUPING OF INDUSTRIAL PRODUCTS

The chapter considers the description of the task of identifying homogeneous batches for the formation of an electronic component base for space applications (as an example of the actual task of automatic grouping with increased requirements for accuracy and stability of the result). Also, it studies the development of a procedure for compiling optimal ensembles of automatic grouping algorithms with the combined use of a genetic algorithm the method of greedy heuristics and a consistent matrix of binary partitions, which makes it possible to increase the accuracy of separation into homogeneous batches of products for practical problems of automatic grouping of industrial products using the approach described in Chapters 2 and 3 to the development of automatic grouping algorithms.

4.1. Problem statement of identifying homogeneous batches of industrial products

One of the most important components of the task of improving the reliability of the system as a whole is the completion of critical components of the system with a component base with increased quality requirements (for example, in the case of electronic equipment). It is important that they have very similar characteristics (to be homogeneous) to ensure the coordinated operation of the same type elements of the system. The homogeneity of the characteristics of the same system's elements is achieved if these elements were made of the same batch of raw materials in the same production batch. Therefore, it is necessary to use the corresponding components manufactured by separate "special" batches. They have increased quality requirements when completing critical system components with increased quality and reliability requirements.

The problems' solution of automatic grouping with increased requirements for accuracy and stability of the result is relevant due to a wide range of their application, both in cluster analysis problems and directly in practical problems in production where high accuracy of result reproducibility is required (for example, for the task of dividing into homogeneous batches of industrial products with special quality requirements).

Chapter 1 states an example of an actual task of automatic grouping with increased requirements for accuracy and stability of the result was considered. Consider it in more detail.

The issues of predicting failures associated with the occurrence of defects at the stage of production of electrical radio products (ERP) were considered in detail in Russian and foreign scientific works [191-199]. All ongoing works for failures prevent is mainly aimed at identifying and eliminating defects directly at the stage of manufacturing electrical and radio products [11, 200, 201]. Moreover, an important task is to develop methods for monitoring the quality of already released batches of incoming industrial products (with increased quality requirements) for compliance with the declared and actual characteristics based on the results of test tests and predicting fault tolerance, including using a retrospective analysis of previously released batches. It is taken in the example of electrical and radio products and in other fields. First of all, control is necessary when using industrial products of foreign production in the case when it is impossible to directly control the products at the stage of production.

Pay attention that the supplied industrial batches of ERP [202] can be heterogeneous (consisting of several production batches of plates), for example, integrated circuits of the same name, but of different quality categories (“OS”, “VP”, “V(S)”, "Q(B)") [203, 204]. Therefore, it is necessary to be sure that we are dealing with a batch of products made from a single (homogeneous) batch of raw materials or that the spread of parameters will be within the acceptable norm in order to extend the test results to the entire production batch of products. Therefore, the identification of homogeneous production batches from prefabricated batches of products is one of the most important measures during testing in order to avoid errors in quality assessment. It directly affects the life of the onboard equipment of the spacecraft.

It is necessary to carry out the prescribed destructive tests for each production batch, consisting of several different groups (batches) in order to make an informed decision on the acceptability of product quality. Therefore, it is necessary to conduct studies to identify such groups [117, 205, 206]. In the case of ERP, to assess the quality of the component base, the following sampling is proposed:

1) electrical and radio products are made of one crystal batch of plates. Then, there is one sample;

2) Electrical radio products are made from more than one crystal batch of plates. Then it is necessary to estimate the number of homogeneous groups, which will be equal to the number of samples.

Pay attention that the samples are formed after industrial products have passed additional screening tests and destructive physical analysis, as a result of which products with potential defects are eliminated.

Thus, automatic grouping of production batches of electrical and radio products is important for ensuring reliability, and in particular radiation resistance [202]. It mainly determines the period of active existence of spacecraft [207, 208].

Thus, automatic grouping of production batches of electrical and radio products is important for ensuring reliability, and in particular radiation resistance [202]. It largely determines the period of active existence of spacecraft [207, 208]. The characteristics of the products included in the special batch should be better than the products of the usual batch of electronic components (even the quality categories "VP" or "OS"), as well as the characteristics of the entire set of products included in the special batch should differ for the better. Thus, a special batch is actually a prototype of the component base of a space quality level.

As it turned out [209], foreign-made electrical and radio products of the "Space" and "Military" quality categories have two differences, i.e., the control of the presence of foreign particles in the under-hull space and the assessment of the drift of parameters during electrical thermal training of products.

When there is no production of special batches of products with increased quality requirements, perhaps the only way out is their formation in specialized test centers using cluster analysis methods with increased requirements for the accuracy and stability of the result (reproducibility of the result of separation into homogeneous batches of industrial products) [210].

The situation with the division into homogeneous batches of industrial products in the process control system for the production of anodes is in some way similar to the process of separation of production batches of electrical and radio products described above, but has its own specifics.

One of the key materials in the production of aluminum is the baked anode [211, 212]. The minimum change in the parameters of the anode entails significant fluctuations in the technical and economic parameters of the electrolysis process (the share of the baked anode in the cost of aluminum production is about 15 percent) [213]. Raw materials for the production of anodes are distinguished by the widest range of parameters of properties that determine the quality of products. Poor quality anodes not only increase aluminum production costs (up to $170 per ton), but also increase greenhouse gas emissions. Consequently, the improvement of the production process for the production of anodes gives great economic prospects for the enterprise [213, 214].

The baked anode quality control system used at a particular enterprise will be considered effective only if it fully simulates the operation of the anode in an aluminum electrolyzer (it is sufficiently sensitive to changes in the properties of the anode). The more complete and better the analysis of raw material parameters is, the more reliable the result is.

In anodes, for example, there may be structural defects (for example, cracks) formed at the stage of forming "green" blocks or under poor firing conditions. Green anode (Eng. Green Anode) is an anode of an aluminum electrolytic bath that has not undergone firing. Since anodes are subjected to severe thermal attack in electrolyzers, their resistance to cracking is of great importance. Failure of an anode in an electrolytic cell due to cracking leads to serious undesirable side effects, which can result in serious losses. Therefore, the task of separating batches of green anodes prior to the firing process is one of the most important in the production of aluminum [211, 212, 214].

The algorithms proposed in Chapters 2 and 3 help to improve the accuracy of automatic grouping methods with increased requirements for accuracy and stability of the result. They can become the basis of an automated system for identifying groups of any industrial products with different parameters.

4.2. Application of search algorithms with alternating neighborhoods for industrial products with high quality requirements

The conceptual diagram of the system for separating prefabricated batches of industrial products (using the example of electrical and radio products for space use) based on the results of test tests conducted at JSC "Testing Technical Center - NPO PM" (JSC "ITC - NPO PM") is presented in Figure 4.1 [210].

The diagram shows the tasks, models, algorithms and their possible relationships that can be involved in building an efficient system for automatic grouping of electrical and radio products into homogeneous production batches [109, 142].

Previously [215], it was shown that the problem of identifying homogeneous batches of industrial products can be reduced to the problem of cluster analysis, where each group (cluster) will represent a homogeneous batch made from one type of raw material. The authors of [216–218] proposed the use of the k-means clustering algorithm to solve the problem of identifying homogeneous batches. In [219], a fuzzy clustering method based on the EM algorithm is considered. A model for separating homogeneous production batches based on a mixture of spherical or uncorrelated Gaussian distributions has been proposed [220]. The application of genetic algorithms with greedy heuristics, as well as modifications of the EM algorithm for separating homogeneous batches of products is considered in [215].

The initial data on the results of tests of industrial products is a multidimensional set of product parameters measured from the results of several hundred non-destructive tests [221]. There were attempts to apply the methods of factor analysis in order to reduce the dimension of input data for clustering products by homogeneous batches [222-224].

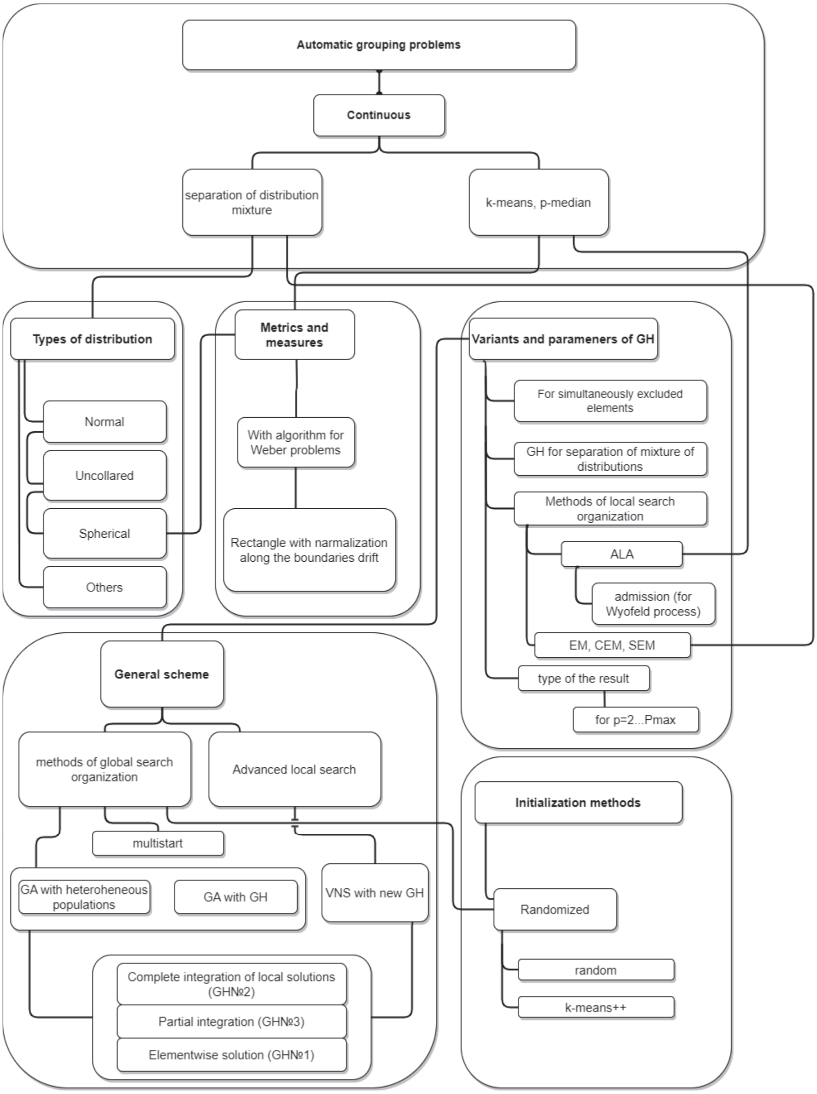

Automatic grouping problems
Continuous
separation of distribution mixture
k-means, p-median
Types of distribution
Normal
Uncollared
Spherical
Others
Metrics and measures
With algorithm for Weber problems
Rectangle with narmalization along the boundaries drift
Variants and parameners of GH
For simultaneously excluded elements
GH for separation of mixture of distributions
Methods of local search organization
ALA
admission (for Wyofeld process)
EM, CEM, SEM
type of the result
for p=2...Pmax
General scheme
methods of global search organization
Advanced local search
multistart
GA with heteroheneous populations
GA with GH
VNS with new GH
Complete integration of local solutions (GHN№2)
Partial integration (GHN№3)
Elementwise solution (GHN№1)
Initialization methods
Randomized
random
k-means++

Fig. 4.1. Conceptual diagram of the system for separating prefabricated batches of industrial products with increased quality requirements based on the results of tests [109]

The studies present that some selected factors depend on the number of considered products in the sample, as well as on the input measured parameters of the product in this sample [222, 223]. However, it has not been possible to single out the optimal universal set of factors for dividing a combined batch consisting of an arbitrary number of homogeneous batches. Thus, despite the fact that factor analysis methods can somewhat reduce the dimension of data, nevertheless, the use of a data array of a sufficiently large dimension is necessary when using cluster analysis methods to separate the combined batch (data remain multidimensional).

One of the problems in data clustering is the automatic determination of the number of clusters (groups). In most cases, the problem of estimating the number of clusters is reduced to the problem of choosing a model. As a rule, automatic grouping algorithms are run in some acceptable limit of the number of possible groups, and the best value (number of clusters) is chosen based on the compactness criterion.

There exist the following main criteria for determining the number of clusters in cluster analysis. They are the Kalinsky-Harabasz index [225], the Davies-Bouldin index (DBI) [226], the Krzanowski-Lai index [227], the Hartigan criterion [228], the Bayes information criterion (BIC, i.e., Bayesian Information Criterion) [229], GAP-criterion [230], Akaike information criterion (AIC) [231], silhouette criterion [232].

Experiments were conducted on the application of each of the listed criteria for industrial products on the example of the results of non-destructive test tests of prefabricated production batches of products [109, 142]. The silhouette criterion turned out to be the most informative and at the same time does not require adjustment of the values of any parameters. In fact, in the production of special batches of IRP, only the silhouette criterion is involved, although in the software implementation of the algorithm for automatic grouping of electrical and radio products by production batches, the results are evaluated according to the criteria of intracluster distance, silhouette, and the Bayesian information criterion.

Application of the silhouette criterion gave the least number of errors in determining the number of production batches [142]. The silhouette criterion also serves to validate the results of automatic grouping, both by lot and by item.

Thus, the use of the silhouette criterion makes it possible to determine effectively the number of production batches in a combined batch. This, in turn, makes it possible to increase the efficiency of the automatic grouping algorithm and more accurately divide into homogeneous batches of electrical radio products.

Automatic grouping models are not universal. Each algorithm has its own field of application. It is necessary to use not one specific algorithm in the case when the field under consideration contains different types of data sets. However, a set of different algorithms to select clusters, since it is not known in advance which of the algorithms will show the best result on a particular data set (or batches of industrial products). The study presented that the algorithms within the same approach showed different results for different tasks. The ensemble (collective) approach makes it possible to reduce the dependence of the final solution on the chosen parameters of the initial algorithms and obtain a more stable (in terms of reproducibility of the result) solution [187-190].

4.3. Ensembles of clustering algorithms

The authors proposed some statistical and other methods for data mining, including tasks of automatic grouping. However, the development of a technology (a method) suitable for solving the widest possible range of clustering problems remains to be an important problem [190, 233]. For example, the repeated studies proved that the application of ensembles of clustering algorithms helps to conclude that they are comparatively efficient for solving a wide range of problems [190]. After that, we have a question about the method of forming an ensemble. As practice shows, the formation of efficient ensembles is fraught with difficulties, since the choice of algorithms that demonstrate the best results for the formation of an ensemble does not always lead to the formation of an ensemble that gives the best accuracy [189, 234, 235].

There exist two main methods for obtaining an ensemble of algorithms [187, 236, 237]:

1. Computation of a co-occurrence matrix.

2. Finding a consensus partition, i.e., a consistent partition with several solutions available, optimal according to some criterion.

The results obtained by various automatic grouping algorithms are used when forming the final solution.

Consider an example of an ensemble of algorithms [238, 239]. It is a combination of sequential k-means algorithms (each of them offers its own partition) and a hierarchical agglomerative algorithm that combines the obtained solutions using a special mechanism.

At the first step, each algorithm, splits the data into clusters using its own distance metric. Then, the accuracy and weight of the algorithm's opinion in the ensemble are calculated according to the formula:

$$W_i = \frac{Acc_i}{\sum_{i=1}^{L} Acc_i}, \tag{4.1}$$

where Acc_i is accuracy of algorithm *i*, that is, the ratio of the number of correctly clustered objects to the size of the entire sample, and *L* is number of algorithms in the ensemble.

For each resulting partition, a preliminary binary difference matrix of size *n* x *n* (where *n* is number of objects) is compiled. It is necessary to determine whether the partition objects are included in one class. Then, a consistent difference matrix is calculated. Each its element is a weighted sum of the elements of the preliminary matrices (using the weight according to formula 4.1). The matrix obtained in such a way is used as input for the hierarchical agglomerative clustering algorithm. After that, using conventional techniques (such as determining the agglomeration distance jump), one can choose the most appropriate cluster solution [238, 239].

As it was mentioned above, it is necessary to compile a binary similarity/difference matrix for each *L* partition in the ensemble in order to obtain the best partition into clusters:

$$H_i=<h_i(i,j)>,$$

where $h_i(i,j)$ is equal to zero if element *i* and element *j* are in the same cluster, and it is equal to 1 if they are not in one cluster.

The next step in compiling an ensemble of automatic grouping algorithms is compiling a consistent matrix of binary partitions:

$$H^* = \langle h^*(i,j) \rangle, \qquad h^*(i,j) = \sum w_i h_i(i,j),$$

where w_i is the weight of the algorithm. We take the weight equal to the average accuracy of the algorithm applied on test problems.

Genetic algorithms gave high efficiency in constructing ensembles of neural networks [83, 240-244]. They are also used for solving problems of automatic grouping. We applied a genetic algorithm of the greedy heuristic method [150, 190] to form an ensemble of arbitrary algorithms. The choice of this method is due to the fact that the algorithms of this method for practical problems yield results that are difficult to significantly improve by other methods in a comparable time. Moreover, computational experiments showed good results (in terms of the value of the objective function and the stability of these values) for problems of automatic grouping of a large number of objects (hundreds of thousands) and large data vectors.

The accuracy of separate clustering algorithms and their ensembles can be estimated from the available labeled sample, i.e., a sample is required in which the belonging of objects to actual groups is known beforehand.

The accuracy of algorithms and their ensembles will be estimated as follows:

$$Fit^1 = A / N \to \max, \tag{4.2}$$

where A is number of correctly clustered objects; N is total number of objects.

The general scheme of the proposed procedure for compiling optimal ensembles of automatic grouping algorithms with the combined use of the genetic algorithm of the greedy heuristic method and the consistent binary partition matrix for practical problems can be described as follows [189, 190]. The algorithms are represented by the results of their work on m test problems, i.e., binary partition matrices. Labeled data are used at the stage of compiling an ensemble of automatic grouping algorithms. Then the calculations go directly to the combined industrial batch of products (which must be divided into homogeneous batches), using the ensemble of the best algorithms selected in each model.

Algorithm 4.1 The procedure for compiling optimal ensembles of automatic grouping algorithms with the combined use of the genetic algorithm of the greedy heuristic method and the consistent binary partition matrix for practical problems

Given: a set of *m* test problems with labeled data (the actual breakdown of data into groups is known in advance), a set of n clustering algorithms C_i, a population size *q*, the number of algorithms in the ensemble *p*.

Solutions ("individuals") in the algorithm are subsets *S* of the clustering algorithms of power *p* selected for the ensemble.

Step 1. Randomly generate q initial solutions, i.e., "individuals" of the algorithm.

Step 2. Evaluate the value of criterion (4.2), averaged over *m* tasks for each individual, an ensemble represented by an "individual", i.e., a set of algorithms for each task. Store the value of the average criterion in the variable Fit_j, where *j* is the number of the "individual".

Step 3. Check STOP conditions (timeout), STOP when conditions are met.

Step 4. Choose randomly with equal probability two numbers of "species" *i, j.* Make the following ensemble: $S = S_i \cup S_j$:

Step 5. For now $|S| > p$ do:

Step 5.1. For each *i* execute: $C_i \in S$:

Step 5.1.1. Exclude the *i*-th algorithm from the ensemble *S*: $S' = S \setminus C_i$.

Step 5.1.2. For S', estimate the value of criterion (4.2) averaged over m tasks by applying the ensemble S' for each task. Store the value of the average criterion in the Fit'_i variable.

Step 5.1.3. Go to the next iteration of the loop 5.1.

Step 5.2. Delete from *S* the algorithm C_i that corresponds to the smallest value Fit'_i. $S' = S \setminus C_i$.

Step 5.3. Next loop iteration 5.

Step 6. For *S*, estimate the value of the criterion (4.2) averaged over *m* tasks, applying the ensemble *S* for each task. Store the value of the averaged criterion in the Fit_{new} variable.

Step 6. Choose the number of "individual" *k* with the smallest value Fit_k. If $Fit_{new} > Fit_k$, then replace the *k*-th individual with *S*. $S_k = S$; $Fit_k = Fit_{new}$.

Go to Step 2.

Figure 4.2 presents a scheme of the procedure for compiling optimal ensembles of automatic grouping algorithms. First, calculations are made by all algorithms for each data set. After that one algorithm of each model is selected. It shows the best indicators of the objective function, and an ensemble of automatic grouping algorithms is already compiled from them. Pay attention that with the help of the genetic algorithm, an ensemble of models is actually compiled, and within the framework of each model; the choice occurs without the participation of the genetic algorithm.

The scheme presents four models of automatic grouping algorithms using the new algorithms described in Chapters 2 and 3. In fact, there can be any number of models (as well as algorithms in each model) (as indicated by the dots in the scheme). Their number depends on the specific problem being solved, computing resources and time available to the researcher (or specialist at a particular enterprise).

The procedure for compiling optimal ensembles of automatic grouping algorithms was used in the implementation of the computer program "System for compiling optimal ensembles of clustering algorithms for the task of identifying production batches of electrical and radio products" (Certificate of state registration of the computer program No. 2019610095 dated 01/09/2019).

We apply this procedure to the problem described above of compiling optimal ensembles of automatic grouping algorithms for separating electrical and radio products by production batches. Genetic algorithms of the greedy heuristics method do not require a large population for their work. We used *q*=10 to compose ensembles of 3 and 5 algorithms (*p*=3, *p*=5).

As test data sets, the results of non-destructive test tests of prefabricated production batches of electrical and radio products were analyzed, carried out at the specialized test center of JSC "ITC - NPO PM" (Zheleznogorsk), to complete the onboard equipment of spacecraft, the composition of which is known in advance [188, 237, 239]. At the same time, prefabricated batches were artificially assembled from several predomos homogeneous batches of electrical and radio products:

- 140УД25АВК is 2 production batches (clusters) and a relatively small amount of data (56 vectors each with a dimension of 18);
- 3OT122A is 2 batches (767 vectors each with dimension 10);
- 1526LE5 is 6 batches (963 vectors each with 41 dimensions).

The task was to split the compiled combined batch into homogeneous components, followed by an analysis of the quality of this splitting.

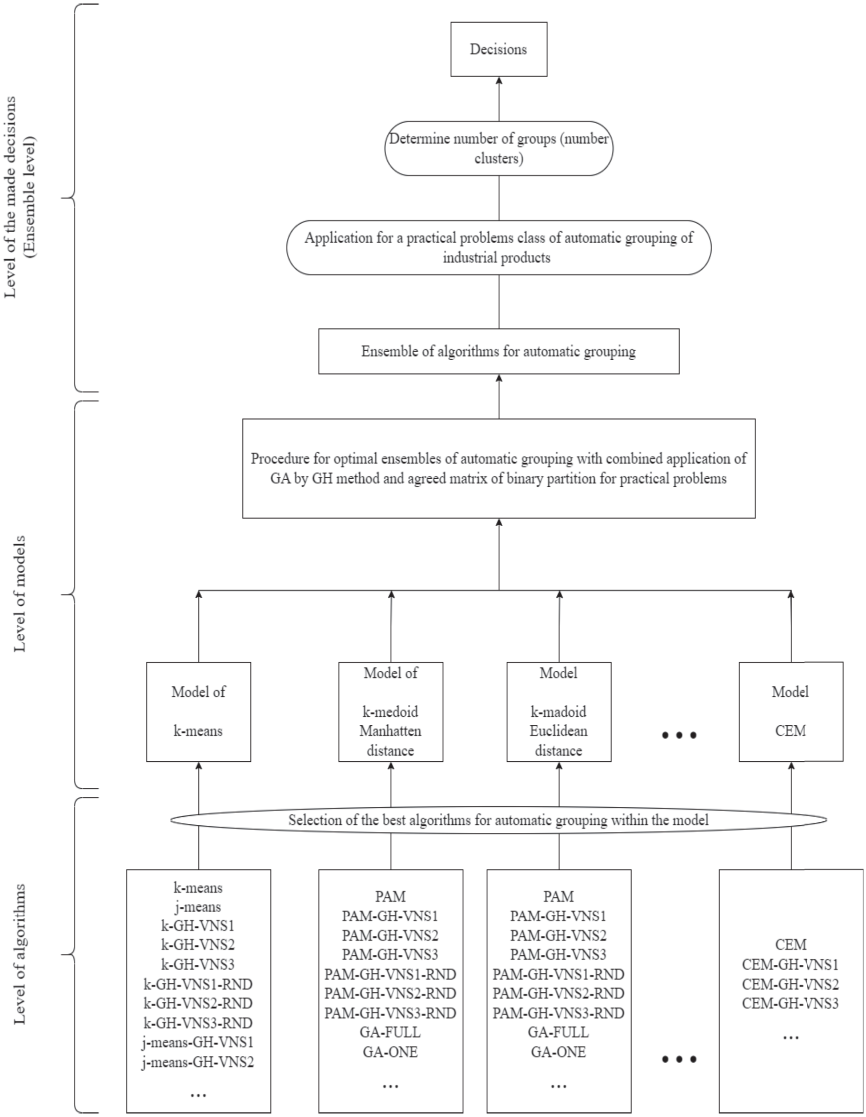

Level of the made decisions (Ensemble level)
Decisions
Determine number of groups (number clusters)
Application for a practical problems class of automatic grouping of industrial products
Ensemble of algorithms for automatic grouping
Level of models
Procedure for optimal ensembles of automatic grouping with combined application of GA by GH method and agreed matrix of binary partition for practical problems
Model of k-means
Model of k-medoid Manhatten distance
Model k-madoid Euclidean distance
• • •
Model CEM
Level of algorithms
Selection of the best algorithms for automatic grouping within the model
k-means
j-means
k-GH-VNS1
k-GH-VNS2
k-GH-VNS3
k-GH-VNS1-RND
k-GH-VNS2-RND
k-GH-VNS3-RND
j-means-GH-VNS1
j-means-GH-VNS2
…
PAM
PAM-GH-VNS1
PAM-GH-VNS2
PAM-GH-VNS3
PAM-GH-VNS1-RND
PAM-GH-VNS2-RND
PAM-GH-VNS3-RND
GA-FULL
GA-ONE
…
PAM
PAM-GH-VNS1
PAM-GH-VNS2
PAM-GH-VNS3
PAM-GH-VNS1-RND
PAM-GH-VNS2-RND
PAM-GH-VNS3-RND
GA-FULL
GA-ONE
…
• • •
CEM
CEM-GH-VNS1
CEM-GH-VNS2
CEM-GH-VNS3
…

Fig. 4.2. Scheme of the procedure for compiling optimal ensembles of automatic grouping algorithms

For research, we used the main classical automatic grouping algorithms [245] for k-means and k-medoid problems, as well as the EM algorithm: k-Means (k-means method) [208, 246-248], k-Means-fast (fast k-means method) [249], k-Means-kernel (k-means kernel method [250], k-Medoids (k-medoid method) [106], EM (Expectation Maximization) [251].

Moreover, to the actual type of the clustering algorithm, the result is significantly affected by the parameters of the algorithms, the values of which can be optimized. By optimization, we mean the selection of such values of the optimized parameters that ensure the maximum accuracy of clustering, that is, the best correspondence of the clustering result to the true division of the combined batch into homogeneous batches of electrical and radio products.

At the output of the process, we evaluate clustering by the Accuracy parameter. By accuracy, we mean the proportion of data objects assigned to the "correct" cluster. This "correctness" can be estimated by having a sample of labeled data, for which their assignment to a particular cluster is known in advance. In this case, our samples are combined from the data of individual homogeneous batches of electrical and radio products. Table 4.1 summarized results are.

Automatic grouping algorithms were used in two implementation options: 1 is classical and 2 is variable. In the second alternative, we are trying to improve the clustering accuracy by changing the variable parameter, i.e., in the k-Means, k-Means(fast) and k-Medoids algorithms, we used the distance measure type. For the k-Means (kernel) algorithm, the kernel type (dot/radial kernel).

According to Table 4.1, clustering algorithms with relatively small amounts of data and the number of production batches (*k* number) show a fairly high accuracy, and with an increase in data volumes and the number of clusters, the clustering accuracy decreases.

At the same time, the most important parameter that affects the result is the distance measure used for automatic grouping models. The use of special measures sometimes makes it possible to adapt simple models like k-means to rather complex clustering problems. In this case, a sufficient condition for the applicability of the distance measure is the existence of an algorithm for solving the corresponding Weber problem, i.e., the problem of finding the center of the cluster [252, 253]. The problem of high computational complexity of some of these algorithms is partially compensated by the parallelization of their execution, shown in Chapter 2.

Table 4.1

Results of computational experiments on production batches of electrical and radio products using separate algorithms for automatic grouping

Algorithm	Accuracy / value of the parameter to be optimized			
	140УД2 5АВК 2 batches	3OT122A 2 batches	1526 LE5 6 batches	1526LE10 7 batches
k-Means-1	100,00 (Euclidean distance)	76,53 (Euclidean distance)	50,57 (Euclidean distance)	39,89 (Euclidean distance)
k-Means (fast)-1	100,00 (Euclidean distance)	67,67 (Euclidean distance)	50,57 (Euclidean distance)	39,89 (Euclidean distance)
k-Means (kernel)-1	100,00 (radial kernel)	59,19 (radial kernel)	47,14 (radial kernel)	46,83 (radial kernel)
k-Medoids-1	100,00 (Euclidean distance)	60,63 (Euclidean distance)	48,60 (Euclidean distance)	37,73 (Euclidean distance)
EM-1	96,43	90,09	no result	no result
k-Means-2	100,00 (Euclidean distance)	76,53 (Euclidean distance)	63,03 (Overlap Similarity)	52,83 (Overlap Similarity)
k-Means (fast)-2	100,00 (Euclidean distance)	76,53 (Euclidean distance)	50,99 (Kernel Euclidean distance)	46,84 (Correlation similarity)
k-Means (kernel)-2	53,57 (dot kernel)	67,67 (dot kernel)	30,22 (dot kernel)	46,83 (dot kernel)
k-Medoids-2	100,00 (Euclidean distance)	91,79 (Euclidean distance)	55,97 (Manhattan distance)	46,83 (Dice Similarity)
EM-2	96,43	95,44	no result	no result

Compose ensembles of three and five, respectively, the best clustering algorithms in terms of accuracy (Table 4.2) for each data set (Table 4.1).

Table 4.2

Results of computational experiments with composed ensembles of clustering algorithms

Production batch / ensemble	140УД 25АВК 2 batches	3OT122A 2 batches	1526LE5 6 batches	1526LE10 7 batches
Ensemble of three algorithms	100,00	95,04	57,01	49,09
Ensemble of five algorithms	100,00	95,44	52,54	47,53

Table 4.3 presents a fragment of calculating the result of an ensemble of five clustering algorithms for the 3OT122A data set.

Table 4.3

A fragment of the results of computational experiments of production batches of 3OT122A electrical and radio products by an ensemble of five clustering algorithms (true and estimated numbers of ERP batches based on clustering results are indicated)

Actual batch	EM-2	k-Medoids-2	EM-1	k-Means-1	k-Means (fast)-2	Ensemble
1	1	2	1	1	1	1
1	1	2	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	2	1	2	2	2
1	1	2	1	1	1	1
1	1	1	1	1	1	1
…	…	…	…	…	…	…
2	2	2	2	1	1	2
2	2	2	2	2	2	2
2	2	2	2	2	2	2
2	2	2	2	1	1	2
…						

We took publicly available and well-known data sets from repositories to formulate conclusions on the results of computational experiments obtained by us with production batches of electrical and radio products for spacecraft and to study the possibility of using ensembles of algorithms for further application:

- Cryotherapy [254, 255] - 2 clusters (90 vectors each with dimension 6);
- pima-indians-diabete - 2 clusters (768 vectors each with dimension 8);
- ionosphere - 2 clusters (351 vectors each with dimension 34);
- Iris - 3 clusters (150 vectors each with dimension 4);
- Zoo - 7 clusters (101 vectors each with dimension 16).

Table 4.4 presents results of the obtained calculations.

Take three and five clustering algorithms, respectively, that showed the best results for each data set (Table 4.4), and compose ensembles of clustering algorithms from them (Table 4.5). Table 4.6 presents ensemble results.

Table 4.4

Results of computational experiments on data sets by individual clustering algorithms

Algorithn	Accuracy / value of the parameter to be optimized				
	Cryotherapy 2 clasters	pima-indians-diabetes 2 кластера	ionosphere 2 clasters	Iris 3 clasters	Zoo 7 clasters
k-Means-1	56,67 (Euclidean Distance)	66,02 (Euclidean Distance)	71,23 (Euclidean Distance)	89,33 (Euclidean Distance)	75,25 (Euclidean Distance)
k-Means (fast)-1	56,67 (Euclidean Distance)	66,02 (Euclidean Distance)	71,23 (Euclidean Distance)	89,33 (Euclidean Distance)	75,25 (Euclidean Distance)
k-Means (kernel)-1	55,56 (radial kernel)	51,17 (radial kernel)	55,56 (radial kernel)	93,33 (radial kernel)	54,46 (radial kernel)
k-Medoids-1	57,78 (Euclidean Distance)	54,43 (Euclidean Distance)	68,09 (Euclidean Distance)	76,67 (Euclidean Distance)	79,21 (Euclidean Distance)

EM-1	56,67	65,62	no result	96,67	no result
k-Means-2	75,56 (CamberraDistance)	66,28 (Manhattan Distance)	no result	96,67 (CosineSimilarity)	83,17 (ManhattanDistance)
k-Means (fast)-2	75,56 (CamberraDistance)	66,28 (Manhattan Distance)	no result	96,67 (CosineSimilarity)	83,17 (ManhattanDistance)
k-Means (kernel)-2	53,33 (dot kernel)	65,10 (dot kernel)	64,10 (dot kernel)	33,33 (dot kernel)	40,59 (dot kernel)
k-Medoids-2	73,33 (CamberraDistance)	66,02 (DynamicTimeWarping Distance)	72,36 (JaccardSimilarity)	97,33 (CosineSimilarity)	80,20 (CosineSimilarity)
EM-2	56,67 (1-st step)	66,28 (1- st step)	no result	96,67 (100-th step)	no result

According to the results of computational experiments, it can be seen that any automatic grouping algorithms for the problem of dividing a combined batch of electrical radio products or a data set from a repository into two homogeneous batches show a fairly high accuracy. With an increase in the number of homogeneous production batches in the combined batch, the accuracy drops. At the same time, for different data sets, the best results are demonstrated by different algorithms.

Table 4.5

Clustering algorithms that gave the best results for each data set

Data set	Cryotherapy 2 clusters	pima-indians-diabetes 2 clusters	ionosphere 2 clusters	Iris 3 clusters	Zoo 7 clusters
1	k-Means-2	k-Means-2	k-Medoids-2	k-Medoids-2	k-Means-2
2	k-Means(fast)-2	k-Means(fast)-2	k-Means-1	EM-1	k-Means (fast)-2
3	k-Medoids-2	EM-2	k-Means (fast)-1	k-Means-2	k-Medoids-2
4	k-Medoids-1	k-Means-1	k-Medoids-1	k-Means (fast)-2	k-Medoids-1
5	EM-1	k-Means(fast)-1	k-Means (kernel)-2	EM-2	k-Means-1

Table 4.6

Results of computational experiments on data sets by ensembles of clustering algorithms

Data set	Cryotherapy 2 clusters	Pima-indians-diabetes 2 clusters	Ionosphere 2 clusters	Iris 3 clusters	Zoo 7 clusters
Ensemble of three algorithms	75,56	66,28	71,23	96,71	83,17
Ensemble of five algorithms	75,56	65,89	68,66	96,67	81,15

The application of the ensemble approach can be more efficient than separate clustering algorithms. Moreover, individual algorithms are able to show results that exceed the results of the ensemble in accuracy, but the accuracy of the ensemble is still higher than the average percentage of the results of separation accuracy using individual algorithms selected for compiling the ensemble on a set of test problems [189, 190, 234, 235].

It is also necessary for a specific task to take into account the number of algorithms used in the ensemble, due to the fact that the accuracy of the ensemble of automatic grouping algorithms for different data sets changes when the number of algorithms in the ensemble changes. Since in practice it is impossible to calculate the accuracy of clustering due to the lack of information about the actual composition of the sample, and it is impossible to predict a priori which of the algorithms in a particular case will show the most adequate results, the use of an ensemble approach to solving such problems is promising and relevant. In particular, the application of the ensemble approach in combination with the new GH-VNS automatic grouping algorithms (discussed in Chapters 2 and 3), which provide the best result within a given model, will allow obtaining results that are not only more adequate, but also reproducible for multiple runs of the algorithm.

4.4. General decision-making scheme for the acceptance of batches of industrial products with increased quality requirements

The conceptual scheme of the system for separating prefabricated batches of industrial products with increased quality requirements was updated based on the results of test tests [109, 142].

The supplemented conceptual scheme (Figure 4.3) presents tasks of automatic grouping, models and algorithms with relationships involved in building an effective system for separating prefabricated batches of industrial products with increased quality requirements for homogeneous production batches.

A software subsystem based on the k-medoid model with various distance measures, as well as a modified greedy heuristic (with partial union) has been added to the existing system for automatic grouping of the separation of prefabricated batches of industrial products into homogeneous production batches based on the k-means model. This approach allows the use of additional competitive mathematical models of automatic grouping to make a decision on the acceptance of lots and combine them into ensembles. One model of automatic grouping makes it possible to verify the results of another model, and if the results do not match under the conditions of the highest requirements for the accuracy and stability of the result of the selection of homogeneous batches, it is proposed to abandon the use of disputed specimens of products when completing the onboard equipment of spacecraft [109, 130].

The data of ongoing test tests, manufacturers of electrical and radio products, product names (nomenclature of tested products), the composition of tests for each product indicating the range of permissible values of each measurement and the test results of each copy in the batch are entered into the database. Each batch of a product in the database is registered with an indication of the possible (intended by the manufacturer) number of production batches in the combined batch. The results of the tests performed are also recorded in the database, after which the automated system analyzes the results and displays their graphical representation.

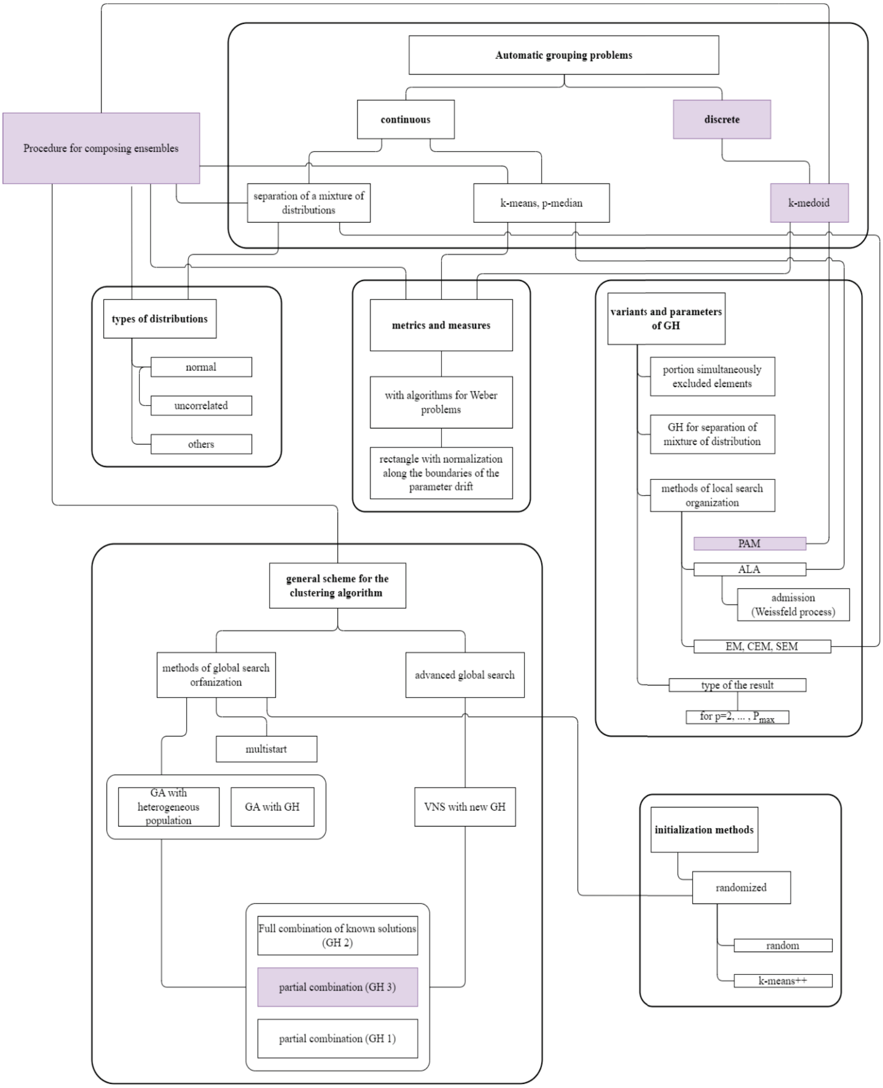

Automatic grouping problems
continuous
discrete
Procedure for composing ensembles
separation of a mixture of distributions
k-means, p-median
k-medoid
types of distributions
normal
uncorrelated
others
metrics and measures
with algorithms for Weber problems
rectangle with normalization along the boundaries of the parameter drift
variants and parameters of GH
portion simultaneously excluded elements
GH for separation of mixture of distribution
methods of local search organization
PAM
ALA
admission (Weissfeld process)
EM, CEM, SEM
type of the result
for p=2, ... , P$_{max}$
general scheme for the clustering algorithm
methods of global search orfanization
advanced global search
multistart
GA with heterogeneous population
GA with GH
VNS with new GH
initialization methods
randomized
random
k-means++
Full combination of known solutions (GH 2)
partial combination (GH 3)
partial combination (GH 1)

Fig. 4.3. The supplemented conceptual scheme of the system for separating prefabricated batches of industrial products with increased quality requirements based on the results of test tests (new components are highlighted in color)

After checking the data prepared by the automated system and the visualization results, the specialist decides whether to accept or reject the production batch of products. For the collection and analysis of statistical information on manufacturers and types of products, data on rejected (with an indication of the reason) lots are also entered into the database.

Some samples are taken from each batch of electrical and radio products, on which a destructive physical analysis is conducted to evaluate the manufacturing process and to evaluate technological defects that are usually not detected at the stage of rejection tests, but appear over time. A special batch of industrial products with an increased quality requirement is obtained according to the results of the tests carried out, after organizing all the necessary and mutually agreed work at the manufacturing plant and at ITC-NPO PM.

* * *

Chapter 4 considers the problem of identifying homogeneous batches for industrial products with increased quality requirements (including for space applications).

The procedure for compiling optimal ensembles of automatic grouping algorithms with the combined use of the genetic algorithm of the greedy heuristic method and the consistent matrix of binary partitions (proposed in this chapter), as well as a new approach to the development of automatic grouping algorithms based on parametric opti- simulation models, with the combined use of search algorithms with alternating randomized neighborhoods and greedy agglomerative heuristic procedures (described in Chapter 3) were used in the development of a system for compiling optimal ensembles of clustering algorithms for the task of identifying production batches and are successfully used in the activities of JSC Testing Technical Center - NPO PM (Zheleznogorsk).

The application of new search algorithms with alternating randomized neighborhoods (including for massively parallel systems) using the above approach and the introduction of a system for compiling optimal ensembles of clustering algorithms for the problem of identifying production batches of industrial products developed as part of the study made it possible to improve the accuracy and stability of the results of assessing the accuracy of separation into homogeneous batches of industrial products, while simultaneously reducing time costs.

Figure 4.4 presents an updated scheme of compatibility of the components of the greedy heuristic method [109] with new components that expand the capabilities of the method for solving problems of automatic grouping.

Earlier, according to the results of research by Stashkov D.V. [109] the scheme was supplemented with one more continuous problem, i.e., separating a mixture of distributions with a block of distribution types.

In the scheme of compatibility of the components of the method of greedy heuristics, as a result of this study, the procedure for compiling optimal ensembles and in the general scheme of the algorithm the subsystem for organizing extended local search, as well as the modified greedy heuristic (with partial union 2) were supplemented (highlighted in lilac). This made it possible to expand the possibilities of the greedy heuristics method for automatic grouping problems with increased requirements for the accuracy and stability of the result.

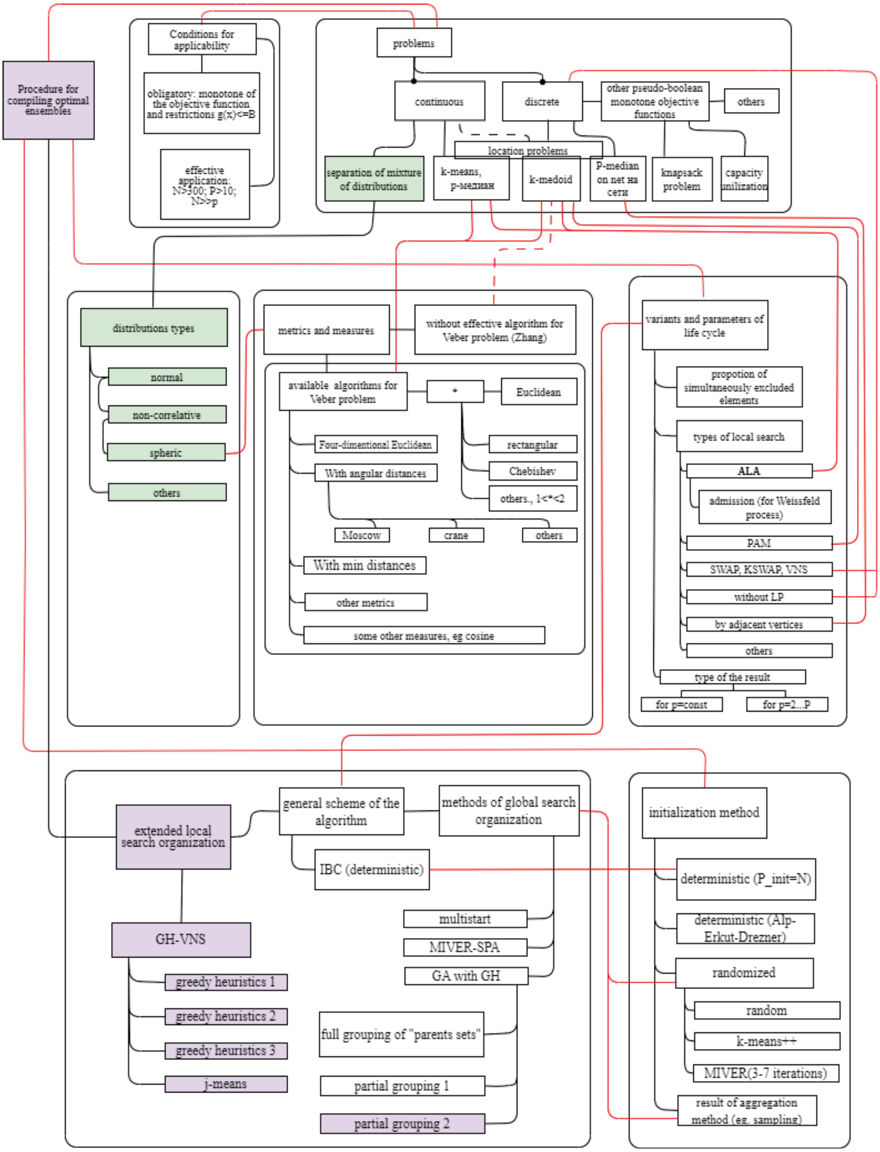

Procedure for compiling optimal ensembles
Conditions for applicability
obligatory: monotone of the objective function and restrictions g(x)<=B
effective application: N>500; P>10; N>>p
problems
continuous
discrete
other pseudo-boolean monotone objective functions
others
location problems
separation of mixture of distributions
k-means, p-медиан
k-medoid
P-median on net на сети
knapsack problem
capacity unilization
distributions types
normal
non-correlative
spheric
others
metrics and measures
without effective algorithm for Veber problem (Zhang)
available algorithms for Veber problem
*
Euclidean
Four-dimentional Euclidean
With angular distances
rectangular
Chebishev
others., 1<*<2
Moscow
crane
others
With min distances
other metrics
some other measures, eg cosine
variants and parameters of life cycle
propotion of simultaneously excluded elements
types of local search
ALA
admission (for Weissfeld process)
PAM
SWAP, KSWAP, VNS
without LP
by adjacent vertices
others
type of the result
for p=const
for p=2...P
extended local search organization
general scheme of the algorithm
methods of global search organization
IBC (deterministic)
multistart
MIVER-SPA
GA with GH
GH-VNS
greedy heuristics 1
greedy heuristics 2
greedy heuristics 3
j-means
full grouping of "parents sets"
partial grouping 1
partial grouping 2
initialization method
deterministic (P_init=N)
deterministic (Alp-Erkut-Drezner)
randomized
random
k-means++
MIVER(3-7 iterations)
result of aggregation method (eg. sampling)

Fig. 4.4. Supplemented compatibility scheme for the components of the method of greedy heuristics

CONCLUSION

The monograph proposes new algorithms for the method of greedy heuristics (including parallel ones) for solving problems of object clustering, combining the use of greedy agglomerative heuristic procedures and extended local search with alternating randomized neighborhoods, allowing solving a range of practical problems with increased accuracy of the result (by the value achieved objective function), as well as the procedure for compiling ensembles of automatic grouping algorithms.

The purpose of the study was achieved by solving the tasks set, namely:

1. The analysis of the existing problems in the application of clustering methods, which are subject to high requirements for the accuracy and stability of the result, revealed a lack of algorithms capable of producing results in a fixed time, which would be difficult to improve by known methods, and which would ensure the stability of the results obtained with multiple runs of the algorithm. At the same time, the well-known algorithms of the greedy heuristic method require significant computational costs.

2. New algorithms for automatic grouping of objects in accordance with the k-means optimization model are developed, based on the joint application of the k-means algorithm, greedy agglomerative heuristic procedures and extended local search with alternating randomized neighborhoods. In this case, the type of the search neighborhood is determined by the type of greedy agglomerative heuristic procedure used, and the randomly generated known solution is a parameter of this neighborhood. It is presented that new algorithms make it possible to obtain a more accurate and stable result (by the achieved value of the objective function) in comparison with the known algorithms, being competitive in comparison with the known algorithms of the greedy heuristics method with a fixed time limit of the algorithm, which allows using algorithms in interactive decision-making mode.

3. New algorithms for automatic grouping of objects based on the k-medoids model, also based on the combined use of greedy agglomerative heuristic procedures, extended local search with alternating randomized neighborhoods, and the Partition around Medoids algorithm, have been developed. It is presented that new algorithms also make it possible to obtain

a more accurate and stable result (in terms of the achieved value of the objective function) in comparison with known algorithms.

4. New algorithms for precise clustering of objects based on a model for separating a mixture of probability distributions using greedy agglomerative heuristic procedures, extended local search with alternating randomized neighborhoods, and a well-known classification EM algorithm are developed, which also have advantages in terms of the value of the objective function obtained in a fixed time. This allows us to speak about a new approach to the development of efficient automatic grouping algorithms based on the combined use of local search algorithms known for the corresponding problems, greedy agglomerative heuristic procedures and search algorithms with alternating randomized neighborhoods formed by applying one of the greedy agglomerative heuristic procedures to the best-known solution and the second solution, generated randomly and being a neighborhood parameter.

5. Parallel modifications of the algorithms of the greedy heuristic method for the CUDA architecture are proposed for the first time. It makes it possible to expand the scope of application of the greedy heuristic method significantly and cover rather large problems up to hundreds of thousands of multidimensional data vectors.

6. A procedure has been developed for compiling optimal ensembles of automatic grouping algorithms with the combined use of the genetic algorithm of the greedy heuristic method and the consistent matrix of binary partitions for practical problems, which makes it possible to reduce the number of errors when dividing a prefabricated batch of industrial products into homogeneous batches using data from non-destructive tests.

REFERENCES

1. Gantz, J.F. The diverse and exploding digital universe. IDC White Paper [Electronic resource]/ J.F. Gantz// Framingham: IDC. – 2008. Access mode: URL http://www.emc.com/collateral/analyst-reports/diverse-exploding-digitaluniverse.pdf (accessed: 01.12.2018).

2. Jain, A.K. Data clustering: 50 years beyond K-means/A.K. Jain// Pattern Recognition Letters.- 2010.- Vol. 31.- P. 651-666.

3. Bolshakova, L.V. Modern mathematical and statistical methods of information processing in scientific and practical work / L.V. Bolshakova, N.A. Yakovleva // Problems of modern science and education. - 2016. - No. 7. Pp. 49-52.

4. Berikov, V.B. Modern trends in cluster analysis [Electronic resource] / V.B. Berikov, G.S. Lbov // All-Russian competitive selection of review and analytical articles in the priority area "Information and telecommunication systems". Novosibirsk: Institute of Mathematics. S.L. Sobolev SB RAS.- 2008.- Pp. 26. Access mode: URL http://www.ict.edu.ru/ft/005638/62315e1-st02.pdf (accessed 21.07.2018).

5. Duda, R. Pattern Classification, second ed./ R. Duda., P. Hart, D. Stork.// New York: John Wiley and Sons.- 2001.- P. 680.

6. Mandel, I.D. Cluster analysis / I.D. Mandel / / M .: Finance and statistics. - 1988. - P. 176.

7. Duk, V.A. Application of data mining technologies in natural sciences, technical and humanitarian areas / V.A. Duke, A.V. Flegontov, I.K. Fomina // Izvestia of the Russian State Pedagogical University im. A.I. Herzen.– 2011.– No. 138.– Pp. 77-84.

8. Tryon, R.C. Cluster analysis/ R.C. Tryon// London: Ann Arbor Edwards Bros. -1939. – P. 139.

9. Vorontsov, K.V. Algorithms for clustering and multidimensional scaling [Electronic resource] / K.V. Vorontsov// Course of lectures.– Moscow State University.– 2007.– Access mode: http://www.ccas.ru/voron/download/Clustering.pdf.

10. Lukyanenko, M.V. Reliability of electronic products in spacecraft equipment: textbook. allowance / M.V. Lukyanenko, N.P. Churlyaeva, V.V. Fedosov // Sib. state aerospace un-ty. - Krasnoyarsk. - 2016. - P. 188.

11. Fedosov, V.V. The minimum required amount of testing of micro-electronics products at the stage of input control / V.V. Fedosov, V.I. Orlov // News of higher educational institutions. Instrumentation. - 2011. - T.54. No. 4.– Pp. 58-62.

12. Iwayama, M. Cluster-based text categorization: A comparison of category search strategies/ M. Iwayama, T. Tokunaga// Proc. 18th ACM Internat. Conf. on Research and Development in Information Retrieval.– 1995.– Pp. 273–281.

13. Barakhnin, V.B. Clustering of text documents based on compound key terms / V.B. Barakhnin, D.A. Tkachev // Vestnik NSU. Series: Information technologies. - 2010. - Vol. 8 / issue. 2.– Pp. 5-14.

14. Barakhnin, V.B. Designing an information system for representing the results of a complex analysis of poetic texts / V.B. Barakhnin, O.Yu. Kozhemyakina, Yu.S. Borzilova // Bulletin of the Novosibirsk State University. Series: Information technologies.- 2019.- Vol. 17. No. 1. Pp. 5-17.

15. Bhatia, S. Conceptual clustering in information retrieval/ S. Bhatia, J. Deogun// IEEE Trans. Systems Man Cybernet.– 1998.– Vol. 28 (B).– Pp. 427–436.

16. Jain, A.K. Image segmentation using clustering/ A.K. Jain, P. Flynn// Advances in Image Understanding.- IEEE Computer Society Press.- 1996.- Pp. 65–83.

17. Arlazarov, V.V. Structural analysis of text fields in systems for streaming input of digitized documents / V.V. Arlazarov, V.M. Klyatskin, O.A. Slavin // Proceedings of the ISA RAS.- 2015.- Vol. 65.- Vol. 1.- Pp. 75-81.

18. Shi, J. Normalized cuts and image segmentation/ J. Shi, J. Malik// IEEE Trans. Pattern Anal. Machine Intell.- 2000.- Vol. 22.- Pp. 888–905.

19. Borisenko, V.I. Image segmentation (state of the problem) / V.I. Borisenko, A.A. Zlatopolsky, I.B. Muchnik // Avtomat. and telemechanics. - 1987. - Vol. 7.- Pp. 3-56.

20. Connell, S.D. Writer adaptation for online handwriting recognition/ S.D. Connell, A.K. Jain// IEEE Trans. Pattern Anal. Machine Intell.- 2002.- Vol. 24.- Issue 3.- Pp. 329–346.

21. Andreeva, E.I. Comparison of digitized pages of business documents based on recognition / E.I. Andreeva, T.V. Manzhikov, O.A. Slavin // Sensory systems. 2018.- Vol. 32. No. 1. Pp. 35-41.

22. Hu, J. Statistical methods for automated generation of service engagement staffing plans/ J. Hu, B.K. Ray, M. Singh// IBM J. Res. Dev.– 2007.– Vol. 51.- Issue 3.– Pp. 281–293.

23. Baldi, P. DNA Microarrays and Gene Expression/ P. Baldi., G. Hatfield.// [s.l.]: Cambridge University Press.- 2002.- Pp.208.

24. Andreev, V.L. Classification constructions in ecology and systematics / V.L. Andreev// M.: Nauka.- 1980.- P. 142.

25. Berry, M.J.A. Data Mining techniques: for marketing, sales, and customer relationship management, 2nd ed./ M.J.A. Berry, G.S. Linoff// [s.l.]: Wiley.– 2004.– P. 464.

26. Galyamov, A.F. Management of interaction with clients of a commercial organization based on the methods of segmentation and clustering of the client base / A.F. Galyamov, S.V. Tarkhov// Vestnik UGATU.– 2014.– Vol. 18.– No. 4(65).– Pp.149-156.

27. Drezner, Z. Facility location: applications and theory/ Z. Drezner, H. Hamacher.// Berlin: Springer-Verlag.– 2004.– P. 460.

28. Farahani, R. Facility location: Concepts, models, algorithms and case studies/ R.Z. Farahani and M. Hekmatfar (eds.)// Berlin Heidelberg: Springer-Verlag.– 2009.– P. 549.

29. Belts, E.A. Optimization of the location of enterprises, taking into account the minimum allowable distances / E.A. Belts, A.A. Kolokolov // Vestnik Omsk un-ty.– 2012.– Vol. 4.– Pp. 13-16.

30. Kochetov, Yu.A. Comparison of metaheuristics for solving the two-level problem of location of enterprises and factory pricing / Yu.A. Kochetov, A.A. Panin, A.V. Plyasunov // Discrete analysis and research of operations.- 2015.- Vol. 22. No 3 (123). Pp. 36-54.

31. Hansen, P. Cluster analysis and mathematical programming/ P. Hansen, B. Jaumard// Mathematical Progralnming.- 1997.- Vol. 79.- Pp. 191-215.

32. Hansen, P. Variable neighborhood search for the p-median/ P. Hansen, N. Mladenovic// Location Science.- 1997.- Vol. 5.- No. 4.- Pp. 207-226.

33. Rosing, R.E. Towards the solution of the (generalized) Weber problem/ R.E. Rosing// Environment and Planning B: Environment and Design.- 1991.- Vol. 18.- Pp. 347-360.

34. Hall, R.W. Median mean and optimum as facility locations/ R.W. Hall// Journal of Regional Science.-1988.- Vol. 28.- Pp. 65-81.

35. Ottaviano, G.I.P. New economic geography: what about the N?/ G.I.P. Ottaviano, J.-F. Thisse// Environment and Planning A.- 2005.- Vol. 37,- Issue 10.- Pp. 1707–1725.

36. Boltyanski, Y. Geometric Methods and Optimization Problems (Combinatorial Optimization)/ Y. Boltyanski, H. Martini, V. Soltan// Dordrecht: Kluwer Academic Publishers.- 1999.- Vol. 4.- P. 432.

37. Volek, J. Location analysis - Possibilities of use in public administration/ J. Volek// Verejna sprava.- Pardubice: Univerzita Pardubice.- 2006.- Pp. 84-85.

38. Teodorovic, D. Transportne mreze, Poglavlje 9: Lokacijski problem/ D. Teodorovic// Beograd: Saobranajni fakultet.- 2009.- Pp. 389-399.

39. Watanabe, D. Generalized Weber Model for Hub Location of Air Cargo/ D. Watanabe, T. Majima, K. Takadama, M. Katuhara// The Eighth International Symposium on Operations Research and Its Applications (ISORA'09).- Zhangjiajie.- 2009.- Pp. 124–131.

40. Beresnev, V.L. Extremal problems of standardization /V.L. Beresnev, E.Kh. Gimadi, V.T. Dementiev // Novosibirsk: Nauka.- 1978.- P. 333.

41. Gimadi, E.Kh. Standardization problem with data of arbitrary sign and connected, quasi-convex and almost quasi-convex matrices / E.Kh. Gimadi // Controlled Systems. Bul. of scientific papers. Vol. 27. - Novosibirsk: Institute of Mathematics of the Siberian Branch of the USSR Academy of Sciences. - 1987. - Pp. 3-11.

42. Gimadi, E.X. Substantiation of a priori estimates of the quality of an approximate solution to the standardization problem / E.Kh. Gimadi // Controlled systems: Sat. scientific Proceedings - Novosibirsk: Institute of Mathematics SO AN USSR. - 1987. - Vol. 27.- Pp. 12-27.

43. Kochetov, Yu.A. Local Search Methods for Discrete Placement Problems: Doctor of Phys.-Math. Sciences: 13/05/18: defended 01/19/2010 / Yu.A. Kochetov // Novosibirsk: Sobolev Institute of Mathematics.- 2010.- P. 259.

44. Vasiliev, I.L. New Lower Bounds for the Placement Problem with customer preferences / I.L. Vasiliev, K.B. Klimentova, Yu.A. Kochetov // Journal of Computational Mathematics and Mathematical Physics. - 2009. - Vol. 49, - No 6.- Pp. 1055-1066.

45. Goncharov, E.N. Behavior of probabilistic greedy algorithms for a multi-stage allocation problem / E.N. Goncharov, Yu.A. Kochetov // Discrete Analysis and Operations Research. Series 2.- 1999.- Vol. 6. No. 1.- Pp. 12-32.

46. Pfeiffer, B. A unified model for Weber problem with continuous and network distance/ B. Pfeiffer, K. Klamroth// Computers and OR. – 2008.– Vol. 35.- No. 2.– Pp. 312-326.

47. Cooper, L. The transportation-location problem/ L. Cooper //Oper. Res.– 1972.– Vol. 20,- No. 1.– Pp. 94-108.

48. Lloyd, S.P. Least Squares Quantization in PCM/ S.P. Lloyd// IEEE Transactions on Information Theory.- 1982.- Vol. 28.- Pp. 129-137.

49. Fermat, P. de Oeuvres/ Fermat P. de (1643), Ed. H.Tannery, ed.// Paris 1891, Supplement: Paris.- 1922.- Vol. 1.- P. 153

50. Torricelli, E. Opere de Evangelista Torricelli/ E. Torricelli, G. Loria, G. Vassura// English edition.– Part 2.– Faenza.– 1919. –Vol I.– Pp. 90-97.

51. Kirszenblat, D. Dubins networks: Thesis/ D. Kirszenblat// Melbourne: Department of Mathematics and Statistics of the University of Melbourne.– 2011.– P. 56.

52. Regional economy and management. Textbook in 2 vol. / Ed. A.I. Gavrilova// N. Novgorod: VVAGS Publishing House.- 2005.- P. 260.

53. Hale, T.S. Location science research: a review/ T.S. Hale, C.R. Moberg// Annals of Operations Research.- 2003.- Vol. 123.- Pp. 21-35.

54. Weiszfeld, E. Sur le point sur lequel la somme des distances de n points donnes est minimum/ E. Weiszfeld// Tohoku Mathematical Journal.– 1937.– Vol. 43.- No. 1.– Pp.335–386.

55. Sturm, R. Ueber den Punkt kleinster Entfernungssumme von gegebenen Punkten/ R. Sturm// J. Rein. Angew. Math.– 1884.– Vol. 97.– Pp. 49–61.

56. Beck, A. Weiszfeld's Method: Old and New Results/ A. Beck, S. Sabach// J. Optim. Theory Appl.– 2015.– Vol. 164,- Iss. 1.– P. 1-40 DOI 10.1007/s10957-014-0586-7.

57. Drezner, Z. The fortified Weiszfeld algorithm for solving the Weber problem/ Z. Drezner// IMA Journal of Management Mathematics.- 2013.- Vol. 26.- P. 1-9. DOI: 10.1093/imaman/dpt019.

58. Hakimi, S.L. Optimum locations of switching centers and the absolute centers and medians of a graph/ L. Hakimi. S.// Operations Research.– 1964.– Vol. 12,- Issue 3.– Pp. 450–459.

59. Hakimi, S.L. Optimum distribution of switching centers in a communication network and some related graph theoretic problems/ S.L. Hakimi // Operations Research.– 1965.– Vol. 13.- No. 3.– Pp. 462–475.

60. Sergienko, I.V. Mathematical models and methods for solving integer optimization problems / I.V. Sergienko // 2nd ed., add. and revised. - Kyiv: Nauko-va Dumka. - 1988. - P. 472.

61. Gimadi, E.Kh. Efficient algorithms for solving a multi-stage location problem on a chain/ E.Kh.Gimadi// Diskretn. analysis and research. Oper.. - 1995. - Volume 2. No. 4. - Pp. 13–31.

62. Alekseev, O.G. Some algorithms for solving the problem of coverage and their experimental verification on a computer / O.G. Alekseev, V.F. Grigoriev // Journal of Computational Mathematics and Mathematical Physics. - 1984. - Vol. 24, - No. 10. - Pp. 1565-1570.

63. Ageev, A.A. Polynomial algorithm for solving the problem of placement on a chain with the same production capacities of enterprises / A.A. Ageev, E.Kh. Gimadi, A.A. Kurochkin // Discrete Analysis and Operations Research.- 2009.- Vol. 16.- No. 5.- Pp. 3–18.

64. Braverman, E.M. Structural methods in the processing of empirical data / E.M. Braverman, I.B. Muchnik// M.: Nauka.- 1983.- P. 464.

65. Zagoruiko, N.G. Competitive similarity as a universal basic tool for cognitive data analysis / N.G. Zagoruiko, I.A. Borisova, O.A. Kutnenko, V.V. Dyubanov, D.A. Levanov // Design ontology. 2015.- Vol. 5. No. 1 (15). Pp. 7-18.

66. Cherenin, V.P. Solution by the method of successive calculations of one class of problems on the location of production / V.P. Cherenin, V.R. Khachaturov // In: Economic and Mathematical Methods, No. 2. - M.: Nauka.- 1965.- Pp. 279-290.

67. Khachaturov, V.R. The Stability of Optimal Values in Problems of Discrete Programming/ V.R. Khachaturov// Optimization Techniques IFIP Technical Conference. Novosibirsk. July 1-7. 1974. Edited by G.I.Marchuk, Springer-Verlag, Berlin, Heidelberg, New York.- 1975.- Pp. 372-376.

68. Cherenin, V.P. Solving some combinatorial problems of optimal planning by the method of sequential calculations / V.P. Cherenin // In the book: Scientific and methodological materials of the economic and mathematical seminar of the LEMM Academy of Sciences of the USSR. Vol 2. - M.: Gipromez. - 1962. - P. 44.

69. Khachaturov, V.R. Algorithms for determining the optimal set of sectoral options for placing enterprises, taking into account the effect of agglomeration / V.R. Khachaturov, N.D. Astakhov, V.V. Grigoriev // M. Computing Center of the Academy of Sciences of the USSR. - 1984. - P. 22.

70. Kolokolov, A.A. Algorithms for decomposition and enumeration of L-classes for solving some placement problems / A.A. Kolokolov, T.V. Levanova // Bulletin of the Omsk University, - 1996. - No. 1. - Pp. 21-23.

71. Levanova, T.V. Local search with alternating neighborhoods for a two-stage location problem / T.V. Levanova, A.S. Fedorenko // Discrete Analysis and Operations Research. 15:3.- 2008.- Pp. 43-57.

72. Kochetov, Y. Large Neighborhood Local Search for the p-Median Problem/ Y. Kochetov E. Alekseeva, T. Levanova, M. Loresh// Yugoslav Journal of Operations Research, 15:1.- 2005. 53–63http://yujor.fon.rs/index.php/journal/ article/view/579/322.

73. Vidyasagar, M. Statistical learning theory and randomized algorithms for control/ M. Vidyasagar// IEEE Control Systems. -1998.- No. 12.- Pp. 69-85.

74. Granichin, O.N. Randomized estimation and optimization algorithms for almost arbitrary noise / O.N. Granichin, B.T. Pole// M. Science. -2003.-P. 291.

75. Goldberg, D.E. Genetic algorithm in search, optimization and machine learning/ D.E. Goldberg// MA: Addison-Wesley.- 1989.- P. 432.

76. Kuehn, A.A. A heuristic program for locating warehouses/ A.A. Kuehn, M.J. Hamburger// Management Science.- 1963.- 9(4).- Pp. 643-666.

77. Kohonen, T. Self-Organization and Associative Memory, 3rd ed./ T. Kohonen// Springer information sciences series.-New York: Springer-Verlag.- 1989.- P. 312.

78. Kirkpatrick, S. Optimization by simulated annealing/ S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi// Science.- 1983.- Vol. 220(4598).- Pp. 671–680.

79. Alp, O. An Efficient Genetic Algorithm for the p-Median Problem/ O. Alp, E. Erkut, Z. Drezner// Annals of Operations Research.- 122 (1-4).- 2003.- P. 21–42. doi 10.1023/A:1026130003508.

80. Chiou, Y. Genetic clustering algorithms/ Y. Chiou, L.W. Lan// European Journal of Operational Research.- 2001.- Vol. 135.- Pp. 413-427.

81. Bozkaya, B.A. Genetic Algorithm for the p-Median Problem/ B. Bozkaya, J. Zhang, E. Erkut// Facility Location: Applications and Theory/ Z. Drezner, H. Hamacher [eds.].-New York: Springer.- 2002.- Pp. 179-205.

82. Krishna, K. Genetic K-means algorithm/ K. Krishna, M. Murty// IEEE Transaction on System, Man and Cybernetics - Part B.- 1999.- Vol. 29.- Pp. 433-439.

83. Holland, J.H. Adaptation in Natural and Artificial System: University of Michigan Press.- 1975.- Pp. 18–25.

84. Reeves, C.R. Genetic algorithms for the operations researcher/ C.R. Reeves// INFORMS Journal of Computing.-1997.- Vol. 9,- Issue 3.- Pp. 231-250.

85. Agarwal, C.C. Optimized crossover for the independent set problem/ C.C. Agarwal, J.B. Orlin, R.P. Tai// Operations research.- 1997.- Vol. 45,- Vol. 2.- Pp. 226-234.

86. Eremeev, A.V. Optimal recombination in genetic algorithms for combinatorial optimization problems, part 1/ A.V. Eremeev, J.V. Kovalenko// Yugoslav Journal of Operations Research.- 2014.- Vol. 24.- No 1.- Pp. 1-20, DOI:10.2298/YJOR130731040E.

87. Steinhaus, H. Sur la division des corps materiels en parties/ H. Steinhaus// Bull. Acad. Polon. Sci.– 1956.– Cl. III,- Vol. IV.– Pp. 801-804.

88. MacQueen, J.B. Some Methods of Classification and Analysis of Multivariate Observations/ J.B. MacQueen// Proceedings of the 5th Berkley Symposium on Mathematical Statistics and Probability.- 1967.- Vol. 1.- Pp. 281-297.

89. Alsabti, K. An efficient k-means clustering algorithm/ K. Alsabti, S. Ranka, V. Singh// Proceedings of IPPS/SPDP Workshop on High Performance Data Mining.- 1998.

90. Nigam, K. Text Classification from Labeled and Unlabeled Documents using EM/ K. Nigam, A.K. Mccallum, S. Thrun, T. Mitchell// ACM journal of Machine Learning-Special issue on information retrieval.- 1999.

91. Kanungo, T. An efficient k-means clustering algorithm: analysis and implementation/ T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu// IEEE Transactions on Pattern Analysis and Machine Intelligence.- 2002.- Vol. 24.- No. 7.- Pp. 881-892.

92. Cheung, Y.M. K-Means: A new generalized k-means clustering algorithm/ Y.M. Cheung // Pattern Recognition Letters.- 2003.- Vol. 24,- Issue 15.- P. 2883-2893.

93. Xiaoli, C. Optimized big data K-means clustering using Map Reduce/ C. Xiaoli [et al.]// Springer Science + Business Media New York.- 2014.

94. Xiong, H. K-Means Clustering Versus Validation Measures: A Data-Distribution Perspective/ H. Xiong, J. Wu, J. Chen// IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.- 2009.- Vol. 39.- No. 2.- Pp. 318-331.

95. Zhang, L. Application of k-means clustering algorithm for classification of NBA guards/ L. Zhang, F. Lu, A. Liu, P. Guo, C. Liu// International Journal of Science and Engineering Applications.- 2016.- Vol. 5.- Issue 1. ISSN- 2319-7560 (online).

96. Wang, J. An Improved K-means Clustering Algorithm/ J. Wang, X. Su// Communication Software and Networks (ICCSN). IEEE 3rd International Conference.- 2011.- Pp. 44-46.

97. Singh, R.V. Data Clustering with Modified K-means Algorithm/ R.V. Singh, M.P.S. Bhatia// Recent Trends in Information Technology. 2011 IEEE International Conference. -2011.- Pp. 717-721.

98. Shi, Na Research on K-means Clustering Algorithm: An Improved K-means Clustering Algorithm/ Shi Na, Liu Xumin, Guan Yong// Intelligent Information Technology and Security Informatics. 2010 IEEE Third International Symposium on 2-4 April,- 2010.- Pp. 63-67.

99. Sharmila Rani, D. Modified K-means Algorithm for Initial Centroid Detection/ D. Sharmila Rani, V.T. Shenbagamuthu// International Journal of Innovative Research in Computer and Communication Engineering.- 2017.- Vol. 2, Special Issue 1.

100. Bhusare, B.B. Initialization for K-Means Clustering using Improved Pillar Algorithm/ B.B. Bhusare, S.M. Bansode Centroids// International Journal of Advanced Research in Computer Engineering & Technology (IJARCET).- 2014.- Vol. 3.- Issue 4.

101. Kaur, K. Statistically Refining the Initial Points for K-Means Clustering Algorithm/ K. Kaur, D. Singh Dhaliwal, R. Kumar Vohra// International Journal of Advanced Research in Computer Engineering & Technology (IJARCET).- 2013.- Vol. 2.- Issue 11.

102. Wang, S. An Improved K-means Clustering Algorithm Based on Dissimilarity/ S. Wang// International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC). Shenyang. China IEEE.- 2013.

103. Mahmud, S. Improvement of K-means clustering algorithm with better initial centroids based on weighted average (англ.)/ S. Mahmud, M. Rahman, N. Akhtar// 7th International Conference on Electrical and Computer Engineering.– IEEE.- 2012-12.– ISBN 9781467314367.– DOI:10.1109/icece.2012.6471633.

104. Abdul Nazeer, K.A. Improving the Accuracy and Efficiency of the k-means Clustering Algorithm/ K.A. Abdul Nazeer, M.P. Sebastian// Proceedings of the World Congress on Engineering.- 2009.- Vol. I. WCE 2009. July 1 – 3.- 2009. London. U.K.

105. Hansen, P. J-Means: a new local search heuristic for minimum sum of squares clustering/ P. Hansen, N. Mladenović// Pattern Recognition.- 2001-02.- Vol. 34.- Issue. 2.- P. 405–413. DOI:10.1016/s0031-3203(99)00216-2.

106. Kaufman, L. Clustering by means of Medoids. Statistical Data Analysis Based on the L1-Norm and Related Methods/ L. Kaufman, P.J. Rousseeuw// Springer US.- 1987.- Pp. 405–416.

107. Korolev, V.Yu. EM-algorithm, its modifications and their application to the problem of separation of mixtures of probability distributions. Theoretical review / V.Yu. Korolev // IPI RAS. M. - 2007. - P. 94.

108. Celeux, G. A classification EM algorithm for clustering and two stochastic versions/ G. Celeux, G. Govaert// Computational Statistics and Data Analysis,- 1992.- Vol. 14.- Pp. 315-332.

109. Kazakovtsev, L.A. Heuristic algorithms for separating a mixture of distributions: monograph /L.A. Kazakovtsev, D.V. Stashkov, V.I. Orlov // under the general editorship of V.I. Orlov; Reshetnev Siberian State University. - Krasnoyarsk, 2018. – P.164.

110. Celeux, G. Classification EM Algorithm for Clustering and Two Stochastic Versions/ G. Celeux, A. Govaert// Rapport de Recherche de l'INRIA RR-1364. Centrede Rocquencourt.- 1991.

111. Broniatowski, M. Reconnaissance de m´elanges de densit´es par un algorithme d'apprentisage probabiliste/ M. Broniatowski, G. Celeux, J. Diebolt // in: E. Diday, M. Jambu, L. Lebart, J.-P. Pag`es, R. Tomasopne (Eds.) Data Analysis and Informatics, III, North Holland, Amsterdam.- 1983.- P. 359-373.

112. Celeux, G. Reconnaissance de m´elanges de densit´e et classification. Un algorithme d'apprentisage probabiliste: l'algorithme SEM/ G. Celeux, J. Diebolt// Rapport de Recherche de l'INRIA RR-0349. Centre de Rocquencourt.- 1984.

113. Celeux, G. The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem/ G. Celeux, J. Diebolt// Computational Statistics Quarterly.- 1985.- Vol. 2.- No. 1.- Pp. 73-82.

114. Likas, A. The global k-means clustering algorithm/ A. Likas, M. Vlassis, J. Verbeek// Pattern Recognition.- 2003.- Vol. 36.- Pp. 451-461.

115. Wu, L.-Y. Capacitated facility location problem with general setup cost/ L.-Y. Wu., X.-S. Zhang, J.-L. Zhang// Computers and Operations Research.- 2006.- Vol. 33.- Pp. 1226–1241.

116. Franca, P.M. An adaptive tabu search algorithm for the capacitated clustering problem/ P.M. Franca, N.M. Sosa, V. Pureza// International Transactions in operational Research.- 1999.- Vol. 6.- Pp. 665–678.

117. Orlov, V.I. On non-parametric diagnostics and process control for the manufacture of electrical and radio products / V.I. Orlov, N.A. Sergeeva.// Vestnik Sib-GAU.- 2013.- Issue. 2(48).- Pp. 70-75.

118. Kuklin, V.I. The results of work on ensuring the quality of domestically produced electrical and radio products for the acquisition of onboard equipment of spacecraft for the period 01.2008–06.2009 / V.I. Kuklin, V.I. Orlov, V.V. Fedosov// VIII Russian Scientific and Technical Conference. Electronic component base of space systems. M.- 2009.- Pp. 64-66.

119. Fedosov, V.V. Technical report. Spacecraft "SESAT" with a period of active operation of 10 years. Principles, methods and results of equipping equipment with electrical and radio products / V.V. Fedosov, V.I. Kuklin, V.I. Orlov, Sh.N. Islyaev et al.// FSUE "Reshetnev NPO PM". - 1999. - P. 408.

120. List of Central Committee-1/96. Electronic products allowed for use in the equipment of the Yamal spacecraft with a 10-year period of active existence / / JSC ETC Cyclone. - 1997. - P. 90.

121. Decision No. SST-TP-97006 on the qualification of electrical and radio products for compliance with the requirements of a spacecraft with a 10-year active life (Edition 1-97) / / Cyclone Engineering and Technology Center JSC. - 1997. - P. 108.

122. Vernova, S.N. Model of near-Earth outer space: In 3 volumes. Vol. 3 Ed. academician. Seventh edition / S.N. Vernova// M.: MSU.– 1983.– P. 133.

123. Resistance of electronic products to the effects of space factors and electrical impulse overloads: a Handbook. T. XII. 4th ed. Vol. 2. Thermal vacuum and electrical effects / / VNII "Elektronstandart". - 1990. - P. 162.

124. Piz, R.L. Radiation testing of semiconductor devices for space electronics / R.L. Piz, A.H. Johnston, J.L. Azarevich / / TIIER. - 1988. - Vol. 76. - No. 11. - Pp. 126-145.

125. Radiation resistance of onboard equipment and elements of space vehicles// I All-Union Scientific and Technical Conference. Conference materials. Tomsk. - 1991. - P. 257.

126. Radiation resistance of materials for radio engineering structures: A Handbook. Ed. N.A. Sidorova, V.K. Knyazev// M.: Soviet radio.– 1976.– P. 567.

127. Malyshev, M.M. Methodology for assessing the radiation reliability of IET under conditions of low-intensity ionizing radiation / M.M.

Malyshev, V.G. Malinin, I.V. Kulikov, Yu.N. Torgashov, M.V. Uzhegov // On Sat. Radiation-reliability characteristics of electronic products in extreme operating conditions. Edited by Yu.N. Torgashova St. Petersburg: Publishing house of the RNII "Elektronstandart". - 1994. - P. 96.

128. Myrova, L.O. Ensuring the resistance of communication equipment to ionizing and electromagnetic radiation. 2nd ed., revised. and additional / L.O. Myrova, A.Z. Chepizhenko// M.: Radio and communication.– 1988.– P. 296.

129. Kononov, V.K. Rejection of potentially unreliable integrated circuits using the radiation-stimulating method / V.K. Kononov, V.G. Malinin, D.A. Ospishchev, V.D. Popov // On Sat. Radiation-reliability characteristics of electronic products under extreme operating conditions. Edited by Yu.N. Torgashova SPb.: Publishing house RNII "Elektronstandart". - 1994.- P. 96.

130. Orlov, V.I. Improved technique for the formation of batches of an electronic component base with special quality requirements / V.I. Orlov, D.V. Stashkov, L.A. Kazakovtsev, I.P. Rozhnov, O.B. Kazakovtseva, I.R. Nasyrov // Modern science-intensive technologies.- 2018.- No. 1.- P. 37-42.

131. Osman, I.H. Metaheuristics: a bibliography/ I.H. Osman, G. Laporte// Ann. Oper. Res.- 1996.- Vol. 63.- Pp. 513-628.

132. Hansen, P. Variable Neighborhood Search/ P. Hansen, N. Mladenovic// Search Methodology/ E.K.Bruke, G.Kendall [eds.].- Springer US.- 2005.- P. 211-238, doi: 10.1007/0-387-28356-0_8.

133. Mladenovic, N. Variable neighborhood search/ N. Mladenovic, P. Hansen// Comput. Oper. Res.- 1997.- Vol. 24.- Pp. 1097-1100.

134. Hansen, P. Variable neighborhood search: principles and applications/ P. Hansen, N. Mladenovic// Eur. J. Oper. Res.- 2001.- Vol. 130.- Pp. 449–467.

135. Brimberg, J. A variable neighborhood algorithm for solving the continuous location-allocation problem/ J. Brimberg, N. Mladenovic// Stud. Locat. Anal.- 1996.- Vol. 10.- Pp. 1-12.

136. Hansen, P. Variable neighborhood decomposition search/ P. Hansen, N. Mladenovic, D. Perez-Brito// J. Heuristics.- 2001.- Vol. 7.- № 4.- Pp. 335-350.

137. Brimberg, J. Improvements and comparison of heuristics for solving the uncapacitated multisource Weber problem/ J. Brimberg, P. Hansen, N. Mladenovic, E. Taillard// Oper. Res.- 2000.- Vol. 48,- № 3.- Pp. 444-460.

138. Lopez, F.G. The parallel variable neighborhood search for the p-median problem/ F.G. Lopez, B.M. Batista, J. Moreno-Perez, M. Moreno-Vega// Res. Rep. Univ. of La Laguna, Spain.- 2000.

139. Kochetov, Yu.A. Local search with alternating neighborhoods / Yu.A. Kochetov, N. Mladenovic, P. Hansen// Diskretn. analysis and research. opera. ser. 2.- 2003.- Vol. 10- No 1.- Pp. 11–43.

140. Goldberg, D.E. Genetic algorithms in search, optimization, and machine learning/ D.E. Goldberg// Reading, MA: Addison-Wesley.- 1989.

141. Kazakovtsev, L. Genetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems/ L.A. Kazakovtsev, A.N. Antamoshkin// Informatica.-2014.- Vol. 38,- No. 3.- Pp. 229-240.

142. Kazakovtsev, L.A. Method of greedy heuristics for systems of automatic grouping of objects: Diss. doc. techn. Sciences / L.A. Kazakovtsev// Krasnoyarsk.- 2016.- P. 429.

143. Semenkin, E.S. The method of generalized adaptive search for optimizing the control of spacecraft: Ph.D. Sciences / E.S. Semenkin // SAA. Krasnoyarsk.- 1997.- P. 400.

144. Korobeinikov, S.P. Methods of multicriteria optimization for problems of synthesis of control of complex objects: dis. cand. techn. sciences / S.P. Korobeinikov// MCC Krasnoyarsk.- 1997.- P. 174.

145. Garipov, V.R. Multi-criteria optimization of control systems for complex objects by methods of evolutionary search: dis... cand. tech. Sciences/ V.R.Garipov// SAA. Krasnoyarsk.- 1999.- P. 138.

146. Semenkin, E.S. On evolutionary algorithms for solving complex optimization problems / E.S. Semenkin, A.V. Gumennikova, M.N. Emelyanova, E.A. Sopov // Vestn. Sib. state aerospace un-ta im. acad. M.F. Reshetnev: Sat. scientific tr. / ed. prof. G.P. Belyakova Sib. state aerospace un-ty. Vol. 5. Krasnoyarsk. - 2003. - Pp. 14-23.

147. Kazakovtsev, L.A. Modification of a genetic algorithm with greedy heuristics for continuous allocation and classification problems / L.A. Kazakovtsev, A.A. Stupina, V.I. Orlov// Control systems and information technologies.- 2014.- Vol. 2(56).- Pp. 35-39.

148. Kazakovtsev, L. Application of algorithms with variable greedy heuristics for k-medoids problems / L. Kazakovtsev, I. Rozhnov // Informatica (Ljubljana). – 2020. – Vol. 44, No. 1. – P. 55-61. – DOI 10.31449/inf.v44i1.2737.

149. Orlov, V.I. Variable neighbourhood search algorithm for k-means clustering / V. I. Orlov, L. A. Kazakovtsev, I. P. Rozhnov [et al.] // IOP Conference Series: Materials Science and Engineering : electronic edition. Vol. 450, Issue 2. – Krasnoyarsk: IOP science, 2018. – P. 022035. DOI:10.1088/1757-899X/450/2/022035.

150. Kazakovtsev, L.A. Fast Deterministic Algorithm for EEE Components Classification/ L.A. Kazakovtsev, A.N. Antamoshkin, I.S. Masich// IOP Conf. Series: Materials Science and Engineering.– 2015.– Vol. 94.– article ID 012015.- P. 10. DOI: 10.1088/1757-899X/04/1012015.

151. Hansen, P. Solving large p-median clustering problems by primal dual variable neighborhood search/ P. Hansen, J. Brimberg, D. Urosevic, N. Mladenovic// Data Mining and Knowledge Discovery.- 2009.- 19,- No. 3.- Pp. 351–375.

152. Hansen, P. Variable neighborhood search for weighted maximum satisfiability problem/ P. Hansen, B. Jaumard, N. Mladenovic, A. Parreira// Les Cahiers du GERAD G-2000-62. Monreal. Canada,- 2000.

153. Rozhnov, I.P. Algorithm for the k-means problem with randomized alternating neighborhoods / I.P. Rozhnov, L.A. Kazakovtsev, M.N. Gudyma, V.L. Kazakovtsev // Control systems and information technologies.- 2018.- No. 3 (73).- Pp. 46-51.

154. Belacel, N. Fuzzy J-Means: a new heuristic for fuzzy clustering/ N. Belacel, P. Hansen, N. Mladenovic// Pattern Recognition.- 2002.- Vol. 35.- Pp. 2193–2200.

155. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml].

156. Clustering basic benchmark [http://cs.joensuu.fi/sipu/datasets].

157. https://www.kdd.org/kdd-cup/view/kdd-cup-2004/Tasks.

158. http://www.machinelearning.ru/wiki/index.php?title=Репозиторий_UCI.

159. Rozhnov, I. P. VNS-Based Algorithms for the Centroid-Based Clustering Problem / I. P. Rozhnov, V. I. Orlov, L. A. Kazakovtsev // Facta

Universitatis, Series: Mathematics and Informatics. – 2019. – Vol. 34, No. 5. – P. 957-972. – DOI 10.22190/FUMI1905957R.

160. https://developer.nvidia.com/cuda-zone.

161. David, Luebke How gpus work/ David Luebke, Greg Humphreys// Computer.- 40(2).- 2007.- Pp. 96–100

162. Zechner, M. Accelerating K-Means on the Graphics Processor via CUDA/ M. Zechner, M. Granitzer.

163. Nguyen Viet Hung Neural network algorithms for solving image coding problems using CUDA technology: Ph.D. tech. Sciences / Nguyen Viet Hung / / Moscow. - 2012. - P. 154.

164. Zheltov, S.A. Efficient Computing in CUDA Architecture in Information Security Applications: Ph.D. tech. Sciences / S.A. Zheltov// Moscow.- 2014.- P. 141.

165. Lutz Efficient k-Means on GPUs/ Lutz, Breß, Zeuch, Markl, Rabl// DaMoN’18. June 11. Houston. TX. USA.- 2018.

166. Rozhnov, I.P. Parallel implementation of the greedy heuristic clustering algorithms / L. A. Kazakovtsev, I. P. Rozhnov, E. A. Popov [et al.] // IOP Conference Series: Materials Science and Engineering : International Workshop "Advanced Technologies in Material Science, Mechanical and Automation Engineering – MIP: Engineering – 2019" / Krasnoyarsk Science and Technology City Hall of the Russian Union of Scientific and Engineering Associations. Vol. 537. – Krasnoyarsk: Institute of Physics and IOP Publishing Limited, 2019. – P. 22052. – DOI 10.1088/1757-899X/537/2/022052.

167. Rozhnov, I.P. Implementation of greedy heuristic clustering algorithms for massively parallel systems / I.P. Rozhnov, V.L. Kazakovtsev// Control systems and information technologies.- 2019.- No. 2 (76).- Pp. 36-40.

168. Struyf, A. Clustering in an Object-Oriented Environment/ A. Struyf, M. Hubert, P. Rousseeuw// Journal of Statistical Software.- 1997.- Issue 1 (4). Pp. 1-30.

169. Kaufman, L. Finding groups in data: an introduction to cluster analysis/ L. Kaufman, P.J. Rousseeuw// New York:Wiley.- 1990.- P. 368.

170. Moreno-Perez, J.A. A Parallel Genetic Algorithm for the Discrete p-Median Problem/ J.A. Moreno-Perez, J.L. Roda Garcia,

J.M. Moreno-Vega// Studies in Location Analysis.- 1994.- Issue 7.- P. 131-141.

171. Wesolowsky, G. The Weber problem: History and perspectives // Location Science.- 1993.- No. 1.- P. 5-23.

172. Drezner, Z. A Trajectory Method for the Optimization of the Multifacility Location Problem with lp Distances/ Z. Drezner, G.O. Wesolowsky// Management Science.- 1978.- Vol. 24.- Pp. 1507–1514.

173. Deza, M.M. Metrics on Normed Structures/ M.M. Deza, E. Deza// Encyclopedia of Distances.- Berlin Heidelberg: Springer.- 2013.- P. 89-99. DOI: 10.1007/978-3-642-30958-85.

174. Nicholson, T. A. J. A sequential method for discrete optimization problems and its application to the assignment, traveling salesman and tree scheduling problems/ T. Nicholson// J. Inst. Math. Appl.- 1965.- Vol. 13.- Pp. 362-375.

175. Page E.S. On Monte Carlo methods in congestion problems. I: Searching for an optimum in discrete situations/ E.S. Page// Oper. Res.- 1965.- Vol. 13,- № 2.- Pp. 291-299.

176. Kernighan, B.W. An efficient heuristic procedure for partitioning graphs/ B.W. Kernighan, S. Lin// Bell Syst. Tech. J.- 1970.- Vol. 49.- Pp. 291-307.

177. Gastrigin, L.A. Random search - specificity, stages of history and prejudices / L.A. Gastrigin // Questions of cybernetics. M.: Nauch. council on the complex problem "Cybernetics" of the Academy of Sciences of the USSR. - 1978. - Vol. 33.- Pp. 3-16.

178. Rozhnov, I.P. Algorithms with alternation of greedy heuristic procedures for discrete clustering problems / I.P. Rozhnov // Control systems and information technologies.- 2019.- No. 1 (75).- Pp. 49-55.

179. Sheng, W. A genetic k-medoids clustering algorithm/ W. Sheng, X. Liu// Journal of Heuristics.- 2006.-Vol. 12,- No. 6.- Pp. 447-466.

180. Cherezov, D.S. Review of the main methods of data classification and clustering / D.S. Cherezov, N.A. Tyukachev // Vestnik Voronezh. state university Ser. System analysis and information technologies. - 2009. - Vol. 2.

181. Kazakovtsev, L. Algorithms with Greedy Heuristic Procedures for Mixture Probability Distribution Separation/ L. Kazakovtsev,

D. Stashkov, M. Gudyma, V. Kazakovtsev// Yugoslav Journal of Operations Research.- 2019.- Vol. 29.- Pp. 51-67.

182. Kazakovtsev, L. Self-adjusting variable neighborhood search algorithm for near-optimal k-means clustering / L. Kazakovtsev, I. Rozhnov, A. Popov, E. Tovbis // Computation. – 2020. – Vol. 8, No. 4. – P. 1-32. – DOI 10.3390/computation8040090.

183. Kazakovtsev, L.A. Improved CEM-algorithm for high-dimensional data / L.A. Kazakovtsev, I.P. Rozhnov, P.F. Shestakov // Science and education: experience, problems, development prospects. Krasnoyarsk. KSAU. - 2019.- Pp. 244-247.

184. Rozhnov, I. Improved Classification EM algorithm for the Problem of Separating Semiconductor Device Production Batches / I. Rozhnov, L. Kazakovtsev, E. Bezhitskaya, S. Bezhitskiy // IOP Conf. Series: MIP: Engineering.- 2019.- Vol. 537.

185. Rozhnov, I.P. Improved classification EM algorithm for the problem of separating semiconductor device production batches / I. Rozhnov, L. Kazakovtsev, E. Bezhitskaya, S. Bezhitskiy // IOP Conference Series: Materials Science and Engineering : International Workshop "Advanced Technologies in Material Science, Mechanical and Automation Engineering – MIP: Engineering – 2019" / Krasnoyarsk Science and Technology City Hall of the Russian Union of Scientific and Engineering Associations. Vol. 537. – Krasnoyarsk: Institute of Physics and IOP Publishing Limited, 2019. – P. 52032. – DOI 10.1088/1757-899X/537/5/052032.

186. Ghosh, J. Cluster ensembles/ J. Ghosh, A. Acharya// Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.- 2011.- Vol. 1(4).- Pp. 305–315.

187. Berikov, V.V. Classification of data using a group of cluster analysis algorithms / V.V. Berikov // Knowledge-Ontology-Theory (ZONT-2015).- 2015.- Pp. 29-38.

188. Rozhnov, I. Ensembles of clustering algorithms for problem of detection of homogeneous production batches of semiconductor devices/ I. Rozhnov, V. Orlov, L. Kazakovtsev// CEUR Workshop Proceedings. OPTA-SCL 2018 - Proceedings of the School-Seminar on Optimization Problems and their Applications. CEUR-WS. -2018.- Vol. 2098.- Pp. 338-348.

189. Rozhnov, I.P. Increase in Accuracy of the Solution of the Problem of Identification of Production Batches of Semiconductor Devices/ I.P. Rozhnov, V.I. Orlov, L.A. Kazakovtsev// 14th International Scientific-Technical Conference on Actual Problems of Electronic Instrument Engineering. APEIE-2018. Volume 1, Part 3. - Pp. 363-367. DOI: 10.1109/APEIE.2018.8546294.

190. Rozhnov, I.P. Compilation of optimal ensembles of clustering algorithms / I.P. Rozhnov, V.I. Orlov, M.N. Gudyma, V.L. Kazakovtsev// Control systems and information technologies.- 2018.- No. 2 (72).- Pp. 31-35.

191. Hamiter, L. The History of Space Quality EEE Parts in the United States/ L. Hamiter// ESA Electronic Components Conference.- Noordwijk. The Netherlands: ESTEC.- 1990.- Nov 12-16.- P. 503-508.

192. Kirkconnell, C.S. High Efficiency Digital Cooler Electronics for Aerospace Applications/ C.S. Kirkconnell, T.T. Luong, L.S. et al. Shaw// Proc. SPIE 9070. Infrared Technology and Applications XL.- Baltimore: SPIE.- 2014.- Article 90702Q.- P. 15. [Electronic resource] Accessed mode DOI:10.1117/12.2053075 (accessed: 01.09.2015).

193. Ooi, M.P.-L. Getting more from the semiconductor test: Data mining with defect-cluster extraction/ M.P.-L. Ooi [et al.]// IEEE Trans.Instrum. Meas.- 2011.- Vol. 60,- No. 10.- Pp. 3300-3317.

194. Kwon, Y. Data mining approaches for modeling complex electronic circuitdesign activities/ Y. Kwon, O.A. Omitaomu, G.-N. Wang// Computer & Industrial Engineering.- 2008.- Vol. 54.- Pp. 229-241.

195. Khamidullina, N.M. Predictions of integrated circuit serviceability in space radiation fields/ N.M. Khamidullina [et al.]// Radiation-Measurements.-1999.- Vol. 30.- Pp. 633-638.

196. Zhao, X. Defect Pattern Recognition on Nano/Micro Integrated Circuits Wafer/ X. Zhao, L. Cui// Proceedings of the 3rdIEEE Int. Conf. on Nano/Micro Engineered and Molecular Systems (Sanya, China,January 6-9, 2008). Sanya, China: [s.n.].- 2008.- Pp. 519-523.

197. Bechow, L. An Improved Method for Automatic Detection and Location of Defects in Electronic Components Using Scanning Ultrasonic Microscopy/ L. Bechow [et al.]// IEEE Transactions on Instrumentation and Measurement.- 2003.- Vol. 52,- No. 1.- Pp. 135-142.

198. Ooi, M.P.-L. Identifying Systematic Failures on Semiconductor Wafers Using ADCAS/ M.P.-L. Ooi [et al.]// Design &Test. IEEE.- 2013.- Vol.30 (5).- Pp. 44-53.

199. Anisimov, V.G. Investigation of complex stacking faults in silicon single crystals / V.G. Anisimov, L.N. Danilchuk, Yu.A. Drozdov [et al.] // Surface. X-ray, synchrotron and neutron studies. - 2004. - No. 11. - Pp. 74-81.

200. Orlov, V.I. Corporate identity: reliability and quality / V.I. Orlov, V.V. Fedosov // Petersburg Journal of Electronics. - 2010. - Vol. 1(62).- Pp. 55-64.

201. Orlov, V.I. On the issue of certification of ERI IP [Electronic resource] / V.I. Orlov, V.V. Fedosov// Scientific and technical seminar "Providing enterprises of the radio-electronic industry with a reliable electronic component base. Issues of import substitution. - M.: CJSC "Testpribor".- 2014.- Access mode: URL http://www.makd.ru/media/downloads/sections/electro/230714/on_the_question_of_certification_esi_ip.pdf (Date of access: 03.05.2015).

202. Fedosov, V.V. Issues of Ensuring the Operability of the Electronic Component Base in Spacecraft Equipment: Proc. Allowance / V.V. Fedosov // Sib. state aerospace Un-ty. - Krasnoyarsk. - 2015. - 68 p.

203. OST B 11 0998-99. Integrated microcircuits. General specifications.

204. MIL-PRF-38535 – Performance Specification: Integrated Circuits (Micricircuit) Manufacturing, General Specifications for. Department of Defence, United States of America. – 2007.

205. Koplyarova, N.V. About non-parametric models in the problem of diagnostics of electrical and radio products. Factory laboratory / N.V. Koplyarova, V.I. Orlov, N.A. Sergeeva, V.V. Fedosov// Diagnostics of materials No. 7.- 2014.- Volume 80.- Pp. 73-77.

206. Kazakovtsev, L.A. A fast deterministic algorithm for classifying an electronic component base according to the criterion of equal reliability / L.A. Kazakovtsev, I.S. Masich, V.I. Orlov, V.V. Fedosov / / Control systems and information technologies. - 2015. - Vol. 4(62). - Pp. 39-44.

207. Danilin, N. Problems of application of a modern industrial electronic component base of foreign production in rocket and space technology / N. Danilin, S. Belosludtsev // Modern electronics.– 2007.– Vol. 7.– Pp. 8-12.

208. Kalashnikov, O.A. Functional control of microprocessors during radiation testing / O.A. Kalashnikov, P.V. Nekrasov, A.A. Demidov // Devices and experimental technique. - 2009. - No. 2. - P. 48.

209. Qualified manufacturers list of products qualified under performance specification MIL-PRF-19500 Semiconductor Devices, General Specification for. Department of Defense.- 2010.- P. 188.

210. Orlov, V.I. An improved method for the formation of production batches of an electronic component base with special quality requirements / V.I. Orlov, D.V. Stashkov., L.A. Kazakovtsev, I.P. Rozhnov, I.R. Nasyrov, O.B. Kazakovtseva // Modern science-intensive technologies.- 2018.- No. 1.- Pp. 37-42.

211. Yanko, E.A. Aluminum production: A manual for craftsmen and workers of the electrolysis workshops of aluminum plants / E.A. Yanko - St. Petersburg, 2007 - 69 p.

212. Savin, A.N. The quality of baked anodes supplied to domestic aluminum plants, their consumption in the process of electrolysis and evaluation of the efficiency of use / A.N. Savin // Tsv. metals. 2007. No. 4. Pp. 84-87.

213. Rozhnov, I. P. Informative Features Selection for Building an Optimization Model of the Aluminum Electrolytic Cell Thermal Regime / I. P. Rozhnov, L. A. Kazakovtsev, M. V. Karaseva // Revista GEINTEC. – 2021. – Vol. 11, No. 4. – P. 605-622. – DOI 10.47059/revistageintec.v11i4.2132.

214. Liner, Yu.A. Promising methods for obtaining aluminum and compounds based on it / Yu.A. Liner, G.A. Milkov, E.N. Samoilov // Tsv. metals. 2012. No. 6. Pp. 42-47.

215. Orlov, V.I. Algorithmic support for decision-making on the selection of microelectronic products for space instrumentation: monograph / V. I. Orlov, L. A. Kazakovtsev, I. S. Masich, D. V. Stashkov / / Sib. state aerospace un-t. - Krasnoyarsk. - 2017. - P. 250.

216. Ackermann, M.R. A Clustering Algorithm for Data Streams J/ M.R. Ackermann et al 2012 K.M. Stream// Exp.Algorithmics 17 2.4:2.1-2.30.

217. Kanungo, T. Computing nearest neighbors for moving points and applications to clustering Proc. of the tenthannual ACM-SIAM symp. on Discrete algorithms/ T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu// Society for Industrial and Applied Mathematics.- 1999.- Pp. 931-932.

218. Kazakovtsev, L.A. Modification of the genetic algorithm with-greedy heuristic for continuous location and classification problems / L.A. Kazakovtsev, A.A. Stupina, V.I. Orlov // Sistemy upravleniya i informatsionnye tekhnologii. - 2(56).- 2014.- Pp. 31-34.

219. Orlov, V.I. Fuzzy clustering of EEE components for space industry/ V.I. Orlov, D.V. Stashkov, L.A. Kazakovtsev, A.A. Stupina// IOP Conference Series: Materials Science and Engineering.- 2016. -Vol. 155. Article ID 012026. http://iopscience.iop.org/article/10.1088/1757-899X/155/1/012026/pdf.

220. Kazakovtsev, L.A. Improved model for detection of homogeneous production batches of electronic components/ L.A. Kazakovtsev, V.I. Orlov, D.V. Stashkov, A.N. Antamoshkin, I.S. Masich// IOP Conference Series: Materials Science and Engineering.- 2017.

221. Kazakovtsev, L.A. Method of greedy heuristics for allocation problems / L.A. Kazakovtsev, A.N. Antamoshkin // Vestnik SibGAU.– 2015.–No. 2.–Pp. 317-325.

222. Shkaberina, G.Sh. Factor analysis using the Spearman matrix in the problem of automatic grouping of electrical and radio products by production batches / G.Sh. Shkaberina, V.I. Orlov, E.M. Tovbis, L.A. Kazakovtsev// Control systems and information technologies, No. 1 (75).- 2019. - Pp. 91-96.

223. Shkaberina, G.Sh. Estimation of the impact of semiconductor device parameters on the accuracy of separating a mixed production batch / G.Sh. Shkaberina, V.I. Orlov, E.M. Tovbis, E.V. Sugak, L.A. Kazakovtsev // IOP Conf. Series: MIP: Engineering.- 2019.- Vol. 537.

224. Uberla, K. Factorenanalyse / K. Uberla // Berlin: Springer-Verlag, -1977. – P. 399.

225. Calinski, T. A dendrite method for cluster analysis / T. Calinski, J. Harabasz // Communications in Statistics.- 1974.- Vol. 3.- P. 1-27. doi: 10.1080/ 03610927408827101.

226. Davies, D.L. A Cluster Separation Measure / D.L. Davies, D.W. Bouldin // IEEE Transactions on Pattern Analysis and Machine Intelligence.- 1979.- PAMI-1 (2).- Pp. 224–227.

227. Krzanowski, W. A criterion for determining the number of groups in a dataset using sum of squares clustering/ W. Krzanowski, Y.Lai// Biometrics.- 1985.- No. 44.- Pp. 23–34.

228. Hartigan, J.A. Clustering Algorithms/ J.A. Hartigan// New York: Wiley.- 1975.- P. 369.

229. Schwarz, G. Estimating the Dimension of a Model/ G. Schwarz// Annals of Statistics. 6.- 1978.- No. 2.- P. 461-464. doi:10.1214/aos/1176344136.

230. Tibshirani, R. Estimating the number of clusters in a data set via the gap statistic/ R. Tibshirani, G. Walther, T. Hastie// Journal of the Royal Statistical Society,- 2001.- Vol. 63.- Pp. 411–423.

231. Akaike, H. A new look at the statistical model identification/ H. Akaike// IEEE Transactions on Automatic Control,- 1974.- Vol. 19 (6).- P. 716–723. doi:10.1109/TAC.1974.1100705.

232. Rousseeuw, P. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis/ P. Rousseeuw// Journal of Computational and Applied Mathematics.- 1987.- Vol. 20.- Pp. 53-65.

233. Loseva, E.D. Algorithm for automated formation of ensembles of neural networks for solving complex problems of data mining / E.D. Loseva, A.N. Antamoshkin // News of TulGU. Technical Sciences.- 2017.- No. 4.- Pp. 234-242.

234. Rozhnov I.P. Scheme of optimal ensembles of clustering algorithms with a combined use of the Greedy Heuristics Method and a matched binary partitioning matrix / I. P. Rozhnov, L. A. Kazakovtsev, A. M. Popov // IOP Conference Series: Earth and Environmental Science / Krasnoyarsk Science and Technology City Hall of the Russian Union of Scientific and Engineering Associations. Vol. 315. – Krasnoyarsk: Institute of Physics and IOP Publishing Limited, 2019. – P. 32031. – DOI 10.1088/1755-1315/315/3/032031.

235. Rozhnov I.P. Ensembles of criteria for determining the number of homogeneous groups in a combined batch of industrial production / I. P. Rozhnov, L. A. Kazakovtsev, M. V. Karaseva [et al.] // IOP Conference Series: Materials Science and Engineering / Krasnoyarsk Science and Technology City Hall of the Russian Union of Scientific and Engineering Associations. Vol. 862. – Krasnoyarsk: Institute of Physics and IOP Publishing Limited, 2020. – P. 42017. – DOI 10.1088/1757-899X/862/4/042017.

236. Rozhnov, I.P. Algorithms of automatic grouping with increased requirements to accuracy and stability of the result / I. P. Rozhnov, L. A. Kazakovtsev, V. I. Orlov, D. L. Mikhnev ; Siberian State University of Science and Technology named after M. F. Reshetnev. M. F. Reshetnev. - Moscow : Infra-M Publishing House, 2020. - 192 c. - (Scientific thought). - ISBN 978-5-16-016641-4.

237. Orlov, V.I. Analysis of clustering algorithms and their ensembles for the problem of identifying production batches of electrical and radio products / V.I. Orlov, I.P. Rozhnov, O.B. Kazakovtseva, L.A. Kazakovtsev// Economics and management of control systems.- 2017.- No. 4.4 (26).- Pp. 486-492.

238. Bochkarev, P.V. Development of an ensemble of clustering algorithms based on changing distance metrics / P.V. Bochkarev, V.S. Kireev // Proceedings of the XVIII International Conference DAMDID/RCDL'2016 "Analytics and Data Management in Data Intensive Domains". Ershovo.- October 11-14, 2016.- Pp. 32-36.

239. Rozhnov, I.P. Ensembles of criteria for determining the number of homogeneous groups in a combined batch of industrial production / I. P. Rozhnov, L. A. Kazakovtsev, M. V. Karaseva [et al.] // IOP Conference Series: Materials Science and Engineering, Krasnoyarsk / Krasnoyarsk Science and Technology City Hall of the Russian Union of Scientific and Engineering Associations. Vol. 862. – Krasnoyarsk: Institute of Physics and IOP Publishing Limited, 2020. – P. 42017. – DOI 10.1088/1757-899X/862/4/042017.

240. Koza, J.R. Genetic Programming/ J.R. Koza// On the Programming of Computers by Means of Natural Selection: MIT Press.- 1992.- Pp. 109 – 120.

241. Huang, J.-J. Two-stage genetic programming (2SGP) for the credit scoring model/ J.-J. Huang, G.-H. Tzeng, Ch.-Sh. Ong// Applied Mathematics and Computation.- 2006.- No.- 174 (2).- Pp. 1039-1053.

242. Integer Magoulas, G.D. Improving the Convergence of the Backpropagation Algorithm Using Learning Rate Adaptation Methods/ G.D. Integer Magoulas, M.N. Vrahatis, G.S. Androulaki// Neural Computation.- 1999.- GR-261.10.- Pp. 1769-1796.

243. Ashish, G. Evolutionary Algorithm for MultiCriterion Optimization: A Survey/ G. Ashish, D. Satchidanada// International Journal of Computing & Information Science.- 2004.- Vol. 2.- No. 1.- Pp. 43- 45.

244. Krutikov, V.N. Research of subgradient methods of training neural networks / V.N. Krutikov, D.V. Aryshev // Bulletin of the Kemerovo State University. 2004. No. 1 (17). Pp. 119-123.

245. Berkhin, P. A survey of clustering data mining techniques/ P. Berkhin// Grouping multidimensional data.- Springer 2006.- Pp. 25-71.

246. MacQueen, J. Some methods for classification and analysis of multivariate observations/ J. MacQueen// Proc. 5th Berkeley Symp. on Math. Statistics and Probability.- 1967.- Pp. 281-297.

247. Bhattacharya, A. A tight lower bound instance for k-means++ in constant dimension/ A. Bhattacharya, R. Jaiswal, N. Ailon// Theory and Applications of Models of Computation. Springer.- 2014.- Pp. 7-22.

248. Arthur, D. How slow is the k-means method? / D. Arthur, S. Vassilvitskii // Proceedings of the twenty-second annual symposium on Computational geometry. ACM.- 2006.- Pp. 144-153.

249. Hamerly, G. Accelerating Lloyd's algorithm for k-means clustering/ G. Hamerly, J. Drake// Partitional Clustering Algorithms. Springer.- 2014.- Pp .41-78.

250. Dhillon, I.S. Kernel k-means: spectral clustering and normalized cuts/ I.S. Dhillon, Y. Guan, B. Kulis// Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '04). ACM.- 2004. New York. USA. Pp. 551-556. DOI=http://dx.doi.org/10.1145/1014052.1014118.

251. Dempster, A. Maximum likelihood estimation from incomplete data / A. Dempster, N. Laird, D. Rubin // Journal of the Royal Statistical Society, Series B.- 1977. Vol. 39.- Pp. 1-38.

252. Antamoshkin, A.N. Placement algorithm with Moscow-Karlsruhe metric / A.N. Antamoshkin, L.A. Kazakovtsev// Control systems and information technologies.- 2012.- T. 49.- No. 3.1.- Pp. 111-115.

253. Kazakovtsev, L.A. Algorithm for the placement problem based on the angular distance / L.A. Kazakovtsev // Fundamental Research.- 2012.- No. 9-4.- Pp. 918-923.

254. Khozeimeh, F. An expert system for selecting wart treatment method/ F. Khozeimeh, R. Alizadehsani, M. Roshanzamir, A. Khosravi, P. Layegh, S. Nahavandi// Computers in Biology and Medicine.- 2017.- Vol. 81. 2/1/.- Pp. 167-175.

255. Khozeimeh, F. Intralesional immunotherapy compared to cryotherapy in the treatment of warts/ F. Khozeimeh, F. Jabbari Azad, Y. Mahboubi Oskouei, M. Jafari, S. Tehranian, R. Alizadehsani, et al.// International Journal of Dermatology.- 2017. DOI: 10.1111/ijd.13535.

APPENDIX
Comparative analysis of computational experiments of various algorithms

Tables A.1-A.3 present comparative results of the obtained computational experiments and previously conducted computational experiments on data sets of electrical radio products using various modifications of the genetic algorithm. The authors made a comparison of results of the new algorithms (k-GH-VNS1, k-GH-VNS2, k-GH-VNS3, k-GH-VNS1-RND, k-GH-VNS2-RND, k-GH-VNS3-RND, j- means-GH-VNS1, j-means-GH-VNS2), known algorithms (k-means, j-means) and various modifications of the genetic algorithm based on the value of the objective function.

Prefabricated batches of electrical and radio products were used for the calculations:

- 1526TL1 - 3 batches (1234 data vectors, each with a dimension of 157);
- 2Д522Б - 5 batches (3711 data vectors, each with dimension 10);
- H5503XM1 - 5 batches (3711 data vectors, each with dimension 229).

The following abbreviations and abbreviations were used in Tables A.1-A.3 [142]: GA is genetic algorithm, GH is greedy heuristics, GAGH is genetic with greedy heuristics with a real alphabet, LP is local search, GA FP is genetic algorithm with recombination of bases of a fixed length [179], IBC is Information Bottleneck Clustering, ZhL is multistart greedy heuristics with local search enabled, ALA multistart is multistart ALA procedures.

The best values of the objective function (minimum value, mean value and standard deviation) are highlighted in bold italics.

Table A.1

Results of computational experiments on production batches of 1526TL1 electrical radio products (10 clusters, 1 minute, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
j-means	***43 841,97***	43 843,51	43 842,59	0,4487
k-means	43 842,10	43 844,66	43 843,38	0,8346
k-GH-VNS1	***43 841,97***	43 844,18	43 842,34	0,9000
k-GH-VNS2	***43 841,97***	43 844,18	43 843,46	1,0817
k-GH-VNS3	***43 841,97***	43 842,10	***43 841,99***	0,0424
k-GH-VNS1-RND	no result	no result		
k-GH-VNS2-RND	no result	no result		
k-GH-VNS3-RND	no result	no result		
j-means-GH-VNS1	***43 841,97***	43 841,97	***43 841,97***	***0,0000***
j-means-GH-VNS2	***43 841,97***	43 844,18	43 842,19	0,6971
GAGH +LP	43 842,10	43 845,73	43 843,72	1,3199
GAGH real, σ e =0.25	***43 841,98***	43 844,18	43 842,6	0,6762
GAGH real partially, σ e =0.25	***43 841,98***	43 841,98	***43 841,98***	1,53E-11
GA FP	***43 841,98***	43 842,34	43 842,10	0,0945
GA classical	43 842,10	43 842,88	43 842,44	0,2349
IBC, σ e =0.25	no result	no result		
Determ. GH, σ e=0.25	45 113,56	45 113,56	45 113,56	***0,0000***
Determ. GH, σ e=0.001	45 021,21	45 021,21	45 021,21	***0,0000***
IBC, σ e =0.001	no result	no result		
GH adapt. σ e =0.25	***43 841,98***	43 842,88	43 842,40	0,4508
GH adapt. σ e =0.001	43 842,75	43 844,18	43 843,92	0,5366
GH, σ e =0.25, β=0.5	***43 841,98***	43 842,74	43 842,21	0,2903
GH, σ e =0.25, β=1	***43 841,98***	43 843,78	43 842,49	0,6596
GH, σ e =0.25, β=3	43 842,75	43 843,52	43 843,32	0,3452
GH, σ e =0.001, β=0.5	***43 841,98***	43 842,59	43 842,12	0,2180
GH, σ e =0.001, β=1	43 842,10	43 844,18	43 843,32	0,9767
GH, σ e =0.001, β=3	43 844,18	43 844,18	43 844,18	***0,0000***
GL, σ e =0.25, β=0.5	43 842,74	43 843,52	43 843,09	0,3987
GL, σ e =0.25, β=1	***43 841,98***	43 843,52	43 842,58	0,5319
GL, σ e =0.25, β=3	43 842,74	43 845,40	43 843,29	0,9700
GL, σ e =0.001, β=0.5	***43 841,98***	43 842,94	43 842,51	0,4630
GL, σ e =0.001, β=1	43 842,34	43 844,18	43 842,92	0,5839
GL, σ e =0.001, β=3	43 842,74	45 118,74	44 191,73	596,6553
ALA multistart	***43 841,98***	43 842,74	43 842,36	0,3165

Table A.2

Results of computational experiments on production batches of 2D522B electrical radio products (10 clusters, 1 minute, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
j-means	7 719,98	7 720,74	7 720,36	1,0174
k-means	7 718,57	7 722,91	7 720,74	2,8714
k-GH-VNS1	7 716,88	7 717,18	7 717,03	0,0738
k-GH-VNS2	7 722,32	7 726,42	7 724,37	1,8752
k-GH-VNS3	7 722,81	7 725,22	7 724,51	1,3946
k-GH-VNS1-RND	no result	no result		
k-GH-VNS2-RND	no result	no result		
k-GH-VNS3-RND	no result	no result		
j-means-GH-VNS1	7 717,22	7 721,40	7 719,81	1,7851
j-means-GH-VNS2	7 717,90	7 720,14	7 719,92	1,4016
GAGH +LP	***7 714,13***	7 715,50	***7 714,61***	0,3837
GAGH real, σ e =0.25	***7 714,15***	7 714,77	***7 714,66***	0,1954
GAGH real partially, σ e =0.25	***7 714,15***	7 714,41	***7 714,29***	0,0899
GAFP	***7 714,14***	7 714,29	***7 714,22***	***0,0612***
GA classical	***7 714,14***	7 714,30	***7 714,21***	***0,0678***
IBC, σ e =0.25	no result	no result		
Determ. GH, σ e=0.25	7 902,21	7 902,21	7 902,21	***0,0000***
Determ. GH, σ e=0.001	no result	no result		
IBC, σ e =0.001	no result	no result		
GH adapt. σ e =0.25	***7 714,24***	7 714,81	***7 714,61***	0,2481
GH adapt. σ e =0.001	***7 714,78***	7 725,76	7 717,91	5,3185
GH, σ e =0.25, β=0.5	***7 714,22***	7 714,83	***7 714,64***	0,2521
GH, σ e =0.25, β=1	***7 714,26***	7 714,79	***7 714,61***	0,2266
GH, σ e =0.25, β=3	***7 714,21***	7 714,48	***7 714,29***	0,0851
GH, σ e =0.001, β=0.5	***7 714,34***	7 725,18	7 716,23	3,9519
GH σ e =0.001, β=1	***7 714,77***	7 715,56	***7 714,90***	0,2918
GH σ e =0.001, β=3	***7 714,55***	7 714,77	***7 714,73***	0,0849
GL, σ e =0.25, β=0.5	***7 714,26***	7 727,78	7 716,28	5,0695
GL, σ e =0.25, β=1	***7 714,29***	7 725,63	7 716,01	4,2413
GL, σ e =0.25, β=3	***7 714,29***	7 727,55	7 716,49	4,8981
GL, σ e =0.001, β=0.5	***7 714,14***	7 714,51	***7 714,36***	0,1316
GL, σ e =0.001, β=1	***7 714,28***	7 725,63	7 715,99	4,2505
GL, σ e =0.001, β=3	***7 714,49***	7 731,48	7 722,03	7,2419
ALA multistart	***7 714,14***	7 714,48	***7 714,26***	0,0982

Table A.3

Results of computational experiments on production batches of H5503XM1 electrical radio products (10 clusters, 1 minute, 30 attempts)

Algorithm	Objective function value			
	Min (record)	Max	Mean	Root mean square deviation
j-means	43 675,96	43 681,52	43 678,74	1,4126
k- means	43 675,90	43 684,88	43 679,77	2,8062
k-GH-VNS1	***43 671,89***	43 671,89	***43 671,89***	***0,0000***
k-GH-VNS2	43 672,24	43 674,44	43 673,34	1,0476
k-GH-VNS3	43 672,84	43 675,76	43 674,30	1,5916
k-GH-VNS1-RND	no result	no result		
k-GH-VNS2-RND	no result	no result		
k-GH-VNS3-RND	no result	no result		
j-means-GH-VNS1	***43 671,89***	43 671,89	***43 671,89***	***0,0000***
j-means-GH-VNS2	43 673,14	43 675,56	43 674,35	0,9162
GAGH +LP	43 702,28	43 766,87	43 739,69	20,3107
GAGH real, σ e =0.25	43 678,79	43 693,63	43 687,01	4,5961
GAGH real partially, σ e =0.25	43 675,79	43 686,87	43 680,82	3,3026
GA FP	43 708,14	43 736,26	43 716,26	8,4025
GA classical	43 703,31	43 724,42	43 715,80	6,1660
IBC, σ e =0.25	no result	no result		
Determ. GH, σ e=0.25	43 830,25	43 830,25	43 830,25	***0,0000***
Determ. GH, σ e=0.001	44 573,13	44 573,13	44 573,13	***0,0000***
IBC, σ e =0.001	no result	no result		
GH adapt. σ e =0.25	no result	no result		
GH adapt. σ e =0.001	43 684,45	43 693,51	43 691,02	3,087926
GH, σ e =0.25, β=0.5	43 692,04	43 711,26	43 699,21	6,032778
GH, σ e =0.25, β=1	43 684,45	43 703,25	43 691,72	6,898894
GH, σ e =0.25, β=3	43 680,28	43 700,12	43 688,81	6,230507
GH, σ e =0.001, β=0.5	43 694,11	43 719,47	43 704,39	7,696556
GH, σ e =0.001, β=1	43 684,19	43 703,13	43 691,67	7,368125
GH, σ e =0.001, β=3	43 683,36	43 690,45	43 686,30	2,600117
GH, σ e =0.25, β=0.5	43 705,63	43 733,45	43 717,25	11,08307
GH, σ e =0.25, β=1	43 702,32	43 734,84	43 714,63	12,79796
GH, σ e =0.25, β=3	43 692,50	43 738,93	43 720,10	17,90737
GH, σ e =0.001, β=0.5	43 707,93	43 740,98	43 720,43	10,20816
GH, σ e =0.001, β=1	43 695,14	43 727,59	43 713,57	11,27041
GH, σ e =0.001, β=3	43 703,31	43 760,96	43 723,19	20,38476
ALA multistart	43 701,35	43 753,06	43 735,46	18,04498

ФЗ № 436-ФЗ	Издание не подлежит маркировке в соответствии с п. 1 ч. 2 ст. 1

Научное издание

Рожнов Иван Павлович,
Казаковцев Лев Александрович,
Карасева Маргарита Владимировна

GH-VNS АЛГОРИТМЫ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ

МОНОГРАФИЯ

ООО «Научно-издательский центр ИНФРА-М»
127214, Москва, ул. Полярная, д. 31В, стр. 1
Тел.: (495) 280-15-96, 280-33-86. Факс: (495) 280-36-29
E-mail: books@infra-m.ru http://www.infra-m.ru

Подписано в печать 13.12.2023.
Формат 60×90/16. Бумага офсетная. Гарнитура Petersburg.
Печать цифровая. Усл. печ. л. 10,13.
Тираж 500 экз. Заказ № 00000
ТК 821209-2130665-131223

Отпечатано в типографии ООО «Научно-издательский центр ИНФРА-М»
127214, Москва, ул. Полярная, д. 31В, стр. 1
Тел.: (495) 280-15-96, 280-33-86. Факс: (495) 280-36-29