Yugoslav Journal of Operations Research
29 (2019), Number 1, 51–67
DOI: https://doi.org/10.2298/YJOR171107030K

ALGORITHMS WITH GREEDY HEURISTIC PROCEDURES FOR MIXTURE PROBABILITY DISTRIBUTION SEPARATION

Lev KAZAKOVTSEV
Department of Systems Analysis and Operations Research, Reshetnev University, prosp.Krasnoyarskii Rabochii 31, Krasnoyarsk, 660037, Russian Federation
levklevk@gmail.com

Dmitry STASHKOV
Department of Systems Analysis and Operations Research, Reshetnev University, prosp.Krasnoyarskii Rabochii 31, Krasnoyarsk, 660037, Russian Federation
stashkov@ngs.ru

Mikhail GUDYMA
Department of Systems Analysis and Operations Research, Reshetnev University, prosp.Krasnoyarskii Rabochii 31, Krasnoyarsk, 660037, Russian Federation
darfai04@gmail.com

Vladimir KAZAKOVTSEV
Department of Computer Educational Technologies, ITMO University, Kronverksky Pr. 49, St. Petersburg, 197101, Russian Federation
vokz@bk.ru

Received: November 2017 / Accepted: November 2018

Abstract: For clustering problems based on the model of mixture probability distribution separation, we propose new Variable Neighbourhood Search algorithms (VNS) and evolutionary genetic algorithms (GA) with greedy agglomerative heuristic procedures and compare them with known algorithms. New genetic algorithms implement a global search strategy with the use of a special crossover operator based on greedy agglomerative heuristic procedures in combination with the EM algorithm (Expectation Maximization). In our new VNS algorithms, this combination is used for forming randomized neighbourhoods to search for better solutions. The results of computational experiments made on classical data sets and the testings of production batches of semiconductor devices shipped for the space industry demonstrate that new algorithms allow us to obtain better

results, higher values of the log likelihood objective function, in comparison with the EM algorithm and its modifications.

Keywords: Clustering, Variable Neighbourhood Search, Genetic Algorithm, Greedy Heuristic, Agglomerative Heuristic, Expectation Maximization.

MSC: 65K05, 90C59.

1. INTRODUCTION

The mixture distribution separation problem belongs to the class of clustering problems based on probability distribution densities with unknown parameters. To solve this kind of problems means that we have to find the parameters of these distributions. A criterion (objective function) for finding the required parameters is the log likelihood function. A clustering problem solution implies the use of data from various origins. Data can be generated by different kinds of probability distributions of a random variable. In our work, we assume that the entire data volume is generated by a mixture of several multidimensional distributions, which obey the normal, or near normal, distribution. Our practical data can be well approximated by a normal distribution.

A general formulation of the mixture distribution separation problem is as follows. Suppose that the distribution density on set $X \subset \mathbb{R}^n$ has the form of a mixture of k distributions (we assume that the distributions are normal) [1, 2, 3]:

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

where $\rho_j(x)$ is the density of the jth mixture component, α_j is its prior probability (its "weight" in the mixture).

The problem of mixture distribution separation is to estimate the parameter vector $\Theta = (\theta_1, ..., \theta_k)$ and vector of prior probabilities $A = (\alpha_1, ..., \alpha_k)$ having a set of independent random observations (samples) when k is known and functions $\rho_j(x)$ are known up to their parameters $\theta_1, ..., \theta_k$.

Values of these parameters are determined by solving the problem of maximizing the log likelihood function for given set $\{x_i\}$ of m data vectors:

$$L(\Theta, A) = \ln \prod_{i=1}^{m} \rho(x_i) = \sum_{i=1}^{m} \ln \sum_{j=1}^{k} \alpha_j \rho_j(x_i) \to \max_{\Theta}.$$

An investigation of the properties of mixture probability distributions for modelling new distributions was started in the 1880s by Newcomb [4] and Pearson [5], and continued by Everitt [6], McLachlan [7], etc.

S. Aivazyan et al. [8] systematized the formulation of clustering problems and corresponding algorithms, including the EM algorithm, and also formulated the general extremal clustering problem.

A. G. McKendrick proposed a procedure similar to the EM algorithm in 1926 [9] which was further developed in the 1950-70 (M. J. R. Healey, M. H. Westmacott [10], M. I. Schlesinger [11] et al.) The name of the EM algorithm (Expectation Maximization) was proposed in 1977 by A. Dempster et al. [12]. V. Y. Korolev [2] systematized approaches to the numerical solution of mixture distribution separation problems, proposed special median modifications of the most popular EM algorithm with improved stability in the results, investigated stability properties of these modifications to perturbations in the data, proved the convergence of the SEM (Stochastic EM) algorithm to a stationary distribution, and also proposed criteria for determining the number of mixture components.

In 2014, L. Kazakovtsev and A. Antamoshkin [13, 14] proposed the Greedy Heuristic Method for clustering problems based on the models of the location theory. Their method uses evolutionary approaches. This method is an extendable approach for building systems for solving location and clustering problems and pseudo-Boolean optimization problems. The systems constructed with this method allow obtaining good results (average value of the objective function and stability of these values) for clustering problems with a large number (up to hundreds of thousands) of objects represented by multidimensional data vectors (up to hundreds of dimensions).

In the case of high-dimensional data, application of known methods encounters additional difficulties. For example, biological data sets usually form data arrays of very high dimensionality. At the same time, the number of objects could be rather small for the efficient use of classical algorithms such as the EM algorithm [1]. Dividing a given lot of semiconductor devices shipped for usage in the space industry into a set of homogeneous production batches manufactured from homogeneous raw materials is a clustering problem (unsupervised learning) with rather high dimensionality of data vectors (up to thousands of dimensions) [1]. The number of clusters (homogeneous batches) is unknown, however, this number has known limitations. The problem is even more complicated because of the possible existence of outliers (separate objects that do not belong to one of the distributions such as semiconductor devices manufactured with essential aberrations).

For clustering problems, two approaches can be used:

a. Hard clustering. Each object belongs to only one cluster. In this case, if there is no clear boundary between clusters in the experimental data, the result will not be appropriate (numerical characteristics of belonging to a cluster will be indistinct).

b. Fuzzy clustering. A result is a matrix, and its elements are probabilistic estimations of the association between the objects and the clusters.

In both approaches, classical methods [1] demand that the number of clusters is known in advance.

In this paper, we consider fuzzy clustering based on the mixture probability distribution separation. Problems with Gaussian (normal) distributions and their particular cases (uncorrelated and spherical Gaussians) are in focus because of the nature of the considered practical problems.

The paper is divided into the following sections. In Section 2, we consider

known methods such as the EM algorithm and its modifications. In Section 3, we propose new greedy agglomerative heuristic procedures based on the EM algorithm. In Section 4, we propose new Variable Neighbourhood Search (VNS) algorithms that use these greedy procedures for forming new neighbourhoods. In Section 5, we propose new genetic algorithms which use new greedy procedures as the crossover operator. In Section 6, results of our computational experiments are shown. The advantages and disadvantages of the new algorithms and future research are summarized in Section 7.

2. KNOWN METHODS

The most popular numerical method for solving the mixture distribution separation problems is the EM algorithm [12, 3, 2] and its modifications [15, 16, 17]. The scope of the EM algorithm is much wider than the mixture distribution separation. It includes structural identification problems [18], parameter inference in state-space models [17], statistical inference [19], nonlinear dimensionality reduction [20] and other applications. An important topic of research is the increase in the performance of this algorithm [21]. In our paper, we focus on the increase of its preciseness.

The result of the EM algorithm for mixture distribution separation problems is a set of parameter values of each of the distributions and their prior probabilities.

The EM algorithm for separation of a mixture of k distributions can be described as follows (Algorithm 1, we consider spherical Gaussian distributions as an example). Let $S \subset \mathbb{R}^n$ be our sample data set of m data vectors. The EM algorithm starts with some initial parameter values $\theta = \langle \mu_i^{\langle 0 \rangle}, \sigma_i^{\langle 0 \rangle} \rangle$ (expectation vector of the ith distribution and its standard deviation, respectively) and initial prior probabilities $\alpha_i^{\langle 0 \rangle}$. These parameters are further updated in accordance with the following two-step procedure (here, t is the iteration number).

In case of the uncorrelated Gaussian distributions, the algorithm operates with the standard deviation vector for each cluster. We used an approach with the separation of mixtures of uncorrelated Gaussian distributions with equal standard deviation vectors for all clusters. In this case, σ_j is the jth dimension of the standard deviation vector (equal for all clusters). Its recalculation is performed in the M-step as follows instead of (1):

$$(\sigma_j^{\langle t+1 \rangle})^2 = \sum_{i=1}^{k} \sum_{x \in S} \left\| x - \mu_i^{\langle t+1 \rangle} \right\|^2 p_i^{\langle t+1 \rangle}(x)/(m\alpha_i).$$

Accordingly, in the case of a multidimensional Gaussian distribution, the algorithm operates with complete covariance matrices and the corresponding inverse matrices. The algorithm can be adapted for problems with many kinds of probability distributions.

Algorithm 1 The EM algorithm

repeat

(E step). Let $\tau_i \sim N(\mu_i^{\langle t \rangle}, (\sigma_i^{\langle t \rangle})^2 I_n)$ *be the density of the* i*th distribution:* $\tau_i(x) = (2\pi)^{-n/2} \sigma_i^{-n} \exp(\|x - \mu_i\|^2 / 2\sigma_i^2)$.

for all data vectors $x \in S$

for $i = 1$ **to** k

calculate the conditional probability that x refers to the ith distribution taking into account its current parameters:

$$p_i^{\langle t+1 \rangle}(x) = \alpha_i^{\langle t \rangle} \tau_i(x) / (\sum_j \alpha_j^{\langle t \rangle} \tau_j(x)). \quad (1)$$

end for

end for

(M Step) Distribution parameters are recalculated as follows:

$$\alpha_i^{\langle t+1 \rangle} = \frac{1}{m} \sum_{x \in S} p_i^{\langle t+1 \rangle}(x); \mu_i^{\langle t+1 \rangle} = \frac{\sum_{x \in S} x p_i^{\langle t+1 \rangle}(x)}{m \alpha_i^{\langle t+1 \rangle}}.; \quad (2)$$

$$\sigma_i^{\langle t+1 \rangle 2} = \frac{1}{d} \sum_{x \in S} \left\| x - \mu_i^{\langle t+1 \rangle} \right\|^2 p_i^{\langle t+1 \rangle}(x). \quad (3)$$

until the log likelihood function

$$L^{\langle t+1 \rangle} = \sum_{x \in S} \sum_{i=1}^{k} \ln(\tau_i p_i^{\langle t+1 \rangle}(x)) \quad (4)$$

is unchanged.

3. GREEDY HEURISTIC PROCEDURES FOR MIXTURE DISTRIBUTION SEPARATION PROBLEM

The main idea of this paper is to apply and compare the greedy heuristics approach in combination with genetic global search strategy and the Variable Neighbourhood Search approach to the clustering problem based on mixture distribution separation.

The idea of greedy agglomerative heuristic procedures [13, 14] is the sequential reduction of the number of clusters in the known solution of the problem. Elements of eliminated clusters are redistributed among other clusters. This procedure starts with an infeasible solution with excessive number of clusters. Each time, greedy heuristic procedure removes one or more clusters until the current solution is feasible.

Our greedy agglomerative heuristic procedures contain two steps.

Greedy agglomerative heuristic procedure

Require: two known ("parent") solutions of our problem. These solutions are represented by pairs of sets $\langle \Theta,\ A\rangle$. Set Θ is a set of distributions in the mixture. Each of the distributions is represented by its parameters. The second set A is a set of corresponding prior probabilities.

1: Parent solution sets are merged into a pair of bigger sets. This pair is an intermediate infeasible solution of our problem with an excessive number of distributions.

2: The basic greedy heuristic procedure decreases sequentially the number of distributions. In each iteration, it eliminates such a distribution in a way that its removal gives the least deterioration in the value of the objective function.

The basic greedy heuristic procedure for mixture distribution separation problems is given below:

Algorithm 2 Basic greedy agglomerative heuristic procedure *(for spherical Gaussian distributions)*

Require: initial number of distributions (fuzzy clusters) K, required number of distributions k, $K > k$, initial solution with K distributions represented by a pair of sets of distribution parameters and their weight coefficients $\langle \Theta, A\rangle = \left\langle \left\{ N(\mu_i^{\langle 0\rangle}, (\sigma_i^{\langle 0\rangle})^2 I_n) \right\}, \left\{ \alpha_i^{\langle 0\rangle} \right\}, i = \overline{1,K} \right\rangle$.

Run Algorithm 1 (the EM algorithm) with initial parameters $\langle \Theta, A\rangle$ to obtain the new improved solution $\langle \Theta, A\rangle \leftarrow EM(\langle \Theta, A\rangle)$.

repeat

for all $i' \in \left\{\overline{1,K}\right\}$

Form a pair of truncated sets

$\langle \Theta', A'\rangle \leftarrow \left\langle \Theta \backslash \left\{ N(\mu_{i'}^{\langle 0\rangle}, (\sigma_{i'}^{\langle 0\rangle})^2 I_n) \right\},\quad A \backslash \{\alpha_{i'}^{\langle 0\rangle}\} \right\rangle$.

Run only one iteration of Algorithm 1 with initial parameters $\langle \Theta',\ A'\rangle$. For the obtained solution, calculate the objective function L in accordance with (1)-(4) :

$L_{i'} \leftarrow L(EM_{1iter.}(\langle \Theta',\ A'\rangle))$.

end for

Find index $i'' \leftarrow \arg\max\limits_{i'=\overline{1,k}} L_{i'}$. Form a pair of truncated sets

$\langle \Theta'', A''\rangle \leftarrow \left\langle \Theta \backslash \left\{ N(\mu_{i''}^{\langle 0\rangle}, (\sigma_{i''}^{\langle 0\rangle})^2 I_n) \right\},\quad A \backslash \{\alpha_{i''}^{\langle 0\rangle}\} \right\rangle$.

Run Algorithm 1 with initial parameters represented by this pair:

$\langle \Theta, A\rangle \leftarrow EM(\langle \Theta'', A''\rangle)$. $K \leftarrow |\Theta|$.

until $K = k$.

Depending on the implemented method of merging known solutions of the problem, we offer three options for new heuristic procedures (Algorithms 3 ,4, 5).

The first procedure (Algorithm 3) complements one of the "parent" options for

solving the distribution separation problem represented by pair $\langle \Theta', A' \rangle$ by each of the elements of the second "parent" solution represented by pair of sets $\langle \Theta'', A'' \rangle$ sequentially. For this pair of sets with only one extra distribution, Algorithm 2 runs.

Algorithm 3 Greedy procedure with partial merger #1

Require: pairs of sets $\langle \Theta', A' \rangle = \left\langle \left\{ N(\mu'^{\langle 0 \rangle}_i, (\sigma'^{\langle 0 \rangle}_i)^2 I_n) \right\}, \left\{ \alpha'^{\langle 0 \rangle}_i \right\}, \quad i = \overline{1, K} \right\rangle$
and $\langle \Theta'', A'' \rangle = \left\langle \left\{ N(\mu''^{\langle 0 \rangle}_i, (\sigma''^{\langle 0 \rangle}_i)^2 I_n) \right\}, \left\{ \alpha''^{\langle 0 \rangle}_i \right\}, \quad i = \overline{1, K} \right\rangle$.
$L_{best} \leftarrow -\infty$; $\Theta_{best} \leftarrow \emptyset$; $A_{best} \leftarrow \emptyset$.

for all $i' \in \left\{ \overline{1, k} \right\}$
Element-by-element, merge sets in pairs $\langle \Theta', A' \rangle$ **and** $\langle \Theta'', A'' \rangle$: $\langle \Theta, A \rangle \leftarrow \left\langle \Theta' \bigcup \left\{ N(\mu''^{\langle 0 \rangle}_{i'}, (\sigma''^{\langle 0 \rangle}_{i'})^2 I_n) \right\}, A' \bigcup \{ \alpha'^{\langle 0 \rangle}_{i'} \} \right\rangle$.
Run the basic greedy heuristic procedure (Algorithm 2) with this initial solution $\langle \Theta, A \rangle$:
$\langle \Theta, A \rangle \leftarrow BasicGreedy(\langle \Theta, A \rangle)$.
Calculate the objective function: $L \leftarrow L(\Theta, A)$.
if $L > L_{best}$
$L_{best} \leftarrow L$; $\langle \Theta_{best}, A_{best} \rangle \leftarrow \langle \Theta, A \rangle$.
end if
end for
return $\langle \Theta_{best}, A_{best} \rangle$.

The next version of this procedure (Algorithm 4) is simpler but it demands more computational resources.

Algorithm 4 Greedy procedure with full merger

Require: (see Algorithm 3).
Merge the sets in pairs $\langle \Theta', A' \rangle$ and $\langle \Theta'', A'' \rangle$:

$$\langle \Theta, A \rangle \leftarrow \left\langle \Theta' \bigcup \Theta'', \quad A' \bigcup A'' \right\rangle.$$

Run the basic greedy heuristic procedure (Algorithm 2):
$\langle \Theta, A \rangle \leftarrow BasicGreedy(\langle \Theta, A \rangle)$.
return $\langle \Theta, A \rangle$.

The third option (Algorithm 5) merges the sets partially. The first set is merged with a randomly chosen subset of the second set. This approach gives comparatively good [22] results for solving k-means, k-medoid, and p-median problems [23].

Algorithm 5 Greedy procedure with partial merger #2

Require: (see Algorithm 3).

1: Choose randomly $r \in \overline{\{2, k-1\}}$ with equal probabilities.
2: $L_{best} \leftarrow -\infty$; $\Theta_{best} \leftarrow \emptyset$; $A_{best} \leftarrow \emptyset$.
3: **for** $i = 1$ **to** $k - r$
4: Form a random subset Θ''' of r elements of set Θ'' and a subset A''' of the corresponding elements of set A''.
5: Merge sets $\langle \Theta, A \rangle \leftarrow \langle \Theta' \bigcup \Theta''', \quad A' \bigcup A''' \rangle$.
6: Run Algorithm 2: $\langle \Theta, A \rangle \leftarrow BasicGreedy(\langle \Theta, A \rangle)$; calculate the objective function: $L \leftarrow L(\Theta, A)$.
7: **if** $L > L_{best}$
8: $L_{best} \leftarrow L$; $\langle \Theta_{best}, A_{best} \rangle \leftarrow \langle \Theta, A \rangle$.
9: **end if**
10: **end for**
11: **return** $\langle \Theta_{best}, A_{best} \rangle$.

Unlike k-means and p-median problems [22], for our problem, the very first computational experiments showed that Algorithm 5 is comparatively inefficient. Its efficiency can be slightly improved as follows (see Step 1 of Algorithm 5: improved version).

Step 1 of Algorithm 5: improved version

1: Choose randomly $r' \in [0, 1)$. Calculate $r = [(k/2 - 2)\ r'^2] + 2$. Here, [.] is the integer part.

In this version of Step 1, the expected number of the elements of the second solution $\langle \Theta'', A'' \rangle$ added to the first solution is smaller than in its original version which was designed for the k-means clustering (the greedy agglomerative procedure for the k-means clustering uses less computational resources).

Such heuristic procedures (Algorithms 3, 4 and 5) do not improve the result of the EM algorithm significantly. However, Algorithms 3 and 4 can be used as a part of more complicated and more efficient search strategies such as VNS or genetic algorithms.

4. NEW VARIABLE NEIGHBOURHOOD SEARCH ALGORITHMS

Variable Neighbourhood Search (VNS, see [24, 25, 26]) is a metaheuristic method for solving combinatorial optimization and global optimization problems. The VNS is used for a wide variety of problems [27, 28] including clustering [29].

Special algorithms for solving mixture distribution separation problem proposed in [30] use the idea of Variable Neighbourhood Search in combination with the greedy agglomerative heuristic procedures. These algorithms try to find better solutions in one of the neighbourhoods of a given known solution. To form this neighbourhood, we use greedy agglomerative heuristic procedures. For many

problems, such algorithms allow us to obtain more precise results in comparison with the EM algorithm and its modifications.

For the VNS algorithm, we must define a set of neighbourhoods S_N of some current solution X. We choose a neighbourhood $S \in S_N$ and search for a solution with better value of the objective function. If such a solution has been found, we replace our current solution with the new one and continue our search process. If the solution can not be found in S, we choose another neighbourhood S from the set S_N and try to search for a better solution in S.

For our problems, we propose the VNS algorithm as follows (see Algorithm 6).

Algorithm 6 Variable neighbourhood search for mixture distribution separation

Require: Initial number of neighbourhood $s_{start} \in \{1, 2, 3\}$, initial solution $\langle \Theta, A \rangle$, randomly chosen.
Run the EM-algorithm: $\langle \Theta, A \rangle \leftarrow EM(\langle \Theta, A \rangle)$.
Initialize the current neighbourhood nubmer: $s \leftarrow s_{start}$.
$i \leftarrow 0; j \leftarrow 0$ *(number of iterations with no improved result in the current neighbourhood and number of switched neighbourhoods with no improved result, respectively).*
loop
Run the EM-algorithm with a random initial solution $\langle \Theta', A' \rangle = EM(random)$.
repeat
Depending on the value of s (values 1, 2 or 3 are allowed), run Algorithm 3, 5 or 4 for the initial solutions $\langle \Theta, A \rangle$ and $\langle \Theta', A' \rangle$, respectively. Obtain the resulting solution $\langle \Theta'', A'' \rangle$.
if $L(\Theta'', A'') > L(\Theta, A)$
$\langle \Theta, A \rangle \leftarrow \langle \Theta'', A'' \rangle$; $i \leftarrow 0$; $j \leftarrow 0$.
else
$i \leftarrow i + 1$.
if $i \le i_{\max}$
$i \leftarrow 0; j \leftarrow j + 1, s \leftarrow s + 1$.
if $s > 3$
$s \leftarrow 1$.
end if
if $j > j_{\max}$
return $\langle \Theta, A \rangle$.
end if
end if
end if
until $j \le j_{\max}$.
end loop

There are two important parameters: $i_{\max}$ (number of unsuccessful searches in the current neighbourhood) and $j_{\max}$ (number of unsuccessful neighbourhood

switches). We set $j_{\max} = 2$ and $i_{\max} = 2k$ (here, k is number of distributions in the mixture). Computational experiments show that parameter s_{start} (the number of the initial neighbourhood) is also important. We performed our experiments with every possible value of $s_{start} \in \{1, 2, 3\}$. In Section 6, results of these experiments are called VNS1, VNS2, VNS3, respectively.

5. NEW GENETIC ALGORITHMS WITH GREEDY HEURISTIC

Evolutionary algorithms including genetic ones show high efficiency in solving hard clustering problems based on the k-means and similar models. Solutions in classical genetic algorithms are represented traditionally with L-bit strings. The Greedy Heuristic Method [13] uses genetic algorithms with real alphabet which encode "individuals" (intermediate solutions of the problem being solved) by sets of points in the space $\mathbb{R}^d$. We use a similar approach that encodes the intermediate solutions in a form of sets of real number vectors for solving the mixture distribution separation problems.

In our Algorithm 7, intermediate solutions are represented by pairs of sets $\Theta_l = \left\{ N(\mu_{l,i}, \sigma^2_{l,i} I_n), i = 1, ..., k \right\}$, $l = \overline{1, N_{POP}}$ and $A_l = \left\{ \alpha_{l,i} = 1/k, i = 1, ..., k \right\}$ where N_{POP} is the population size of the the algorithm, i.e. the number of "individuals" (intermediate solutions) used by our algorithm for recombination and generating new intermediate solutions (child solutions). Three versions of Algorithm 7 use various greedy procedures (Algorithms 3-5).

6. COMPUTATIONAL EXPERIMENTS

In our experiments, the execution time for all algorithms was chosen in such a way that at least during the last third of this time, none of the algorithms improved the objective function. The results achieved by the algorithms were fixed during the entire execution time (Fig. 1).

In Table 1, two data sets are arrays of results of the non-destructive tests of the integrated circuits, other data sets in Tables 1 and 2 are classical data sets from the UCI dataset repository [31] and repository of Clustering datasets of the School of Computing of the University of Eastern Finland [32].

For some problems, the new genetic algorithm (GA) allows obtaining the best results in comparison with known algorithms: classical EM algorithm in the multistart mode, CEM algorithm (Classification EM [15]) multistart, SEM algorithm (Stochastic EM [16, 33, 17]) multistart, new Variable Neighbourhood Search algorithm for mixture distribution separation (the table includes results for VNS1, VNS2 and VNS3 described in Section 4). However, the VNS algorithms for several mixture distribution separation problems allow us to obtain better results in a shorter time. Genetic algorithms take more time for obtaining appropriate results. Thus, VNS is more efficient for the largest problem in case of limited time (see KDDCUP Bio 04 data set). Both GA and VNS are inefficient for small-scale problems (see 140UD17AVK tests data set) and data sets with Boolean data (see Chess data set).

Algorithm 7 Genetic algorithm with greedy heuristic. Versions: GA-FULL, GA-ONE, GA-RAND

Require: Population size N_{POP}.

(Initialization Step): Generate randomly N_{POP} initial solutions represented by the sets of the distributions $\Theta_l = \left\{ N(\mu_{l,i}, \sigma_{l,i}^2 I_n), i = 1, ..., k \right\}$, $l = \overline{1, N_{POP}}$ and the corresponding sets of weight coefficients $A_l = \left\{ \alpha_{l,i} = 1/k, i = 1, ..., k \right\}$.

In case of separation of normal distributions, initial values of standard deviation are equal for all distributions and calculated as the standard deviation of the full data set: $\sigma_i^2 = \frac{1}{d} \sum_{x \in S} \|x - \overline{x}\|^2$ *. The values of expectations* $\mu_{l,i}$ *are assigned equal to coordinates of randomly chosen data vectors.*

Run the EM algorithm runs for each of the initial solutions, the obtained objective function values are stored to variables $f_1, ..., f_{N_{POP}}$. Initialize $N_{iter} \leftarrow 0$.

loop

  if the stop conditions are reached

    STOP;

    return the solution with the best (highest) objective function value $f_1, ..., f_{N_{POP}}$ among the population.

  end if *We use a runtime limitation as the stop condition.*

  $Niter \leftarrow Niter + 1$; $N_{POP} \leftarrow \max\{N_{POP}; \lceil \sqrt{1 + N_{iter}} \rceil + 2\}$;

  if N_{POP} has changed

    Initialize solution $\langle \Theta_{N_{POP}}, A_{N_{POP}} \rangle$ as described in the Initialization Step.

  end if

  Select randomly two indexes $k_1, k_2 \in \{\overline{1, N_{POP}}\}, k_1 \neq k_2$. Run Algorithm 4 or Algorithm 3 for the pair of solutions represented by sets Θ_{k_1},Θ_{k_2} and A_{k_1}, A_{k_2}. Store the result to $\langle \Theta', A' \rangle$. *Note: Version GA-FULL of our algorithm runs Algorithm 4, version GA-ONE runs Algorithm 3 and version GA-RAND randomly chooses one of these two algorithms with equal probabilities.*

  Select index $k_3 \in \{\overline{1, N_{POP}}\}$. We use the simplest tournament selection: the algorithm chooses $k_4, k_5 \in \{\overline{1, N_{POP}}\}$ in a random way;

  if $f_{k_4} < f_{k_5}$

    $k_3 = k_4$;

  else

    $k_3 = k_5$.

  end if

  $\langle \Theta_{k_3}, A_{k_3} \rangle \leftarrow \langle \Theta', A' \rangle$; $f_{k_3} \leftarrow L(\Theta', A')$.

end loop

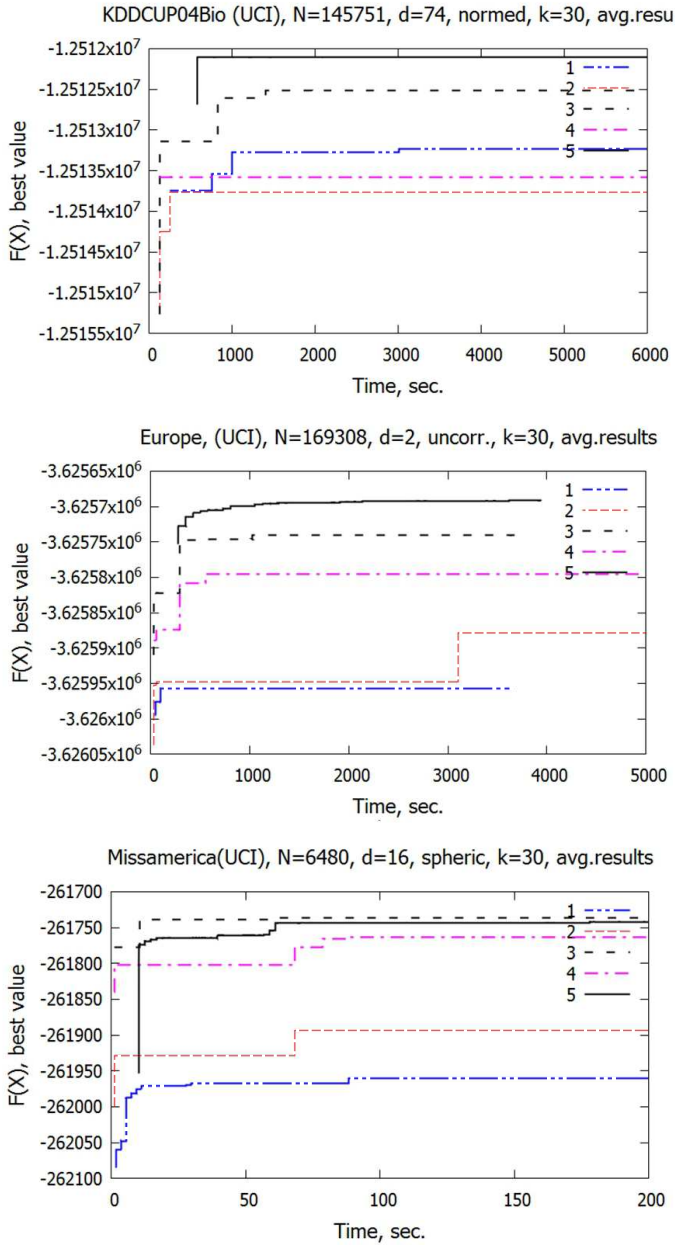

Figure 1: Dynamics of changes in the results of algorithms (1 - EM, 2 - GA-ONE, 3 - GA-FULL, 4 - GA-RAND, 5 -VNS1)

Dataset, number of instances m, dimensionality d	Number of distributions k, type of distributions, time limit	Algorithm	Avg. result $L(\Theta, A)$ (30 runs)	Std. dev. of results
Chess (King-Rook vs. King-Pawn, m=3196, d=36, Boolean	10, spher., 3 min.	EM	-30525.03*	0
		CEM	-30564.13	32.29
		SEM	-30560.70	46.39
		VNS1	-30554.87	28.69
		VNS2	-30539.85	25.43
		VNS3	-30580.60	0
		GA-FULL	-30581.09	1.83
		GA-ONE	-30532.44	19.55
		GA-RAND	-30554.67	28.69
IC tests 140UD17AVK, m=51, d=46	2, uncorrel., 5 sec.,	EM	3790.1*	104.4
		VNS1	3665.630	0
		VNS2	3665.630	0
		VNS3	3673,672	44.043
		GA-FULL	3665.6	0
		GA-ONE	3697.8	83.4
		GA-RAND	3707.7	88.0
IC tests 1526LE2, m=3987, d=206	5, uncorrel., 3 min.	EM	340951.0	1123.3
		GA-FULL	366487.3	968.1
		GA-ONE	350292.0	532.1
		GA-RAND	443491.0*↑⇑	2125.1
KDDCUP04 Bio, m=145751, d=74, normalized	30, spher., 500 min.	EM	-12513230	1256
		CEM	-12512718	343
		SEM	-12514337	684
		VNS1	-12511968*↑⇑	105
		VNS2	-12511984.9	278
		VNS3	-1251265.0	139
		GA-FULL	-12512517	159
		GA-ONE	-12512818	411
		GA-RAND	-12512811	639
Europe, m=169308, d=2	40, spher., 1.5 hours	EM	-3625957.4	49.56
		CEM	-3637892.57	283.94
		SEM	-3625779.08	25.06
		VNS1	-3625694.1	20.148
		VNS2	-3625691.7*↑⇑	14.77
		VNS3	-3625748.7	15.402
		GA-FULL	-3625740.60	15.19
		GA-ONE	-3625878.2	36.34
		GA-RAND	-3625816.2	48.03

Table 1: Comparative results

Dataset, number of instances m, dimensionality d	Number of distributions k, type of distributions, time limit	Algorithm	Avg. result $L(\Theta, A)$ (30 runs)	Std. dev.
Ionosphere (UCI), m=351, d=35, normalized	10, spher., 30 sec.	EM	-871.405	15.79
		CEM	-893.44	7.88
		SEM	-879.95	15.19
		VNS1	-847.463	23.972
		VNS2	-849.352	28.073
		VNS3	-878.153	41.875
		GA-FULL	-960.32	23.50
		GA-ONE	-824.85*↑⇑	4.47
		GA-RAND	-832.83	14.80
Mopsi locations (Joensuu), m=6014, d=2,	20, spher., 40 min.	EM	39268.66	9967.64
		CEM	48424.17	237.30
		SEM	36272.22	9619.60
		VNS1	50291.12	122.86
		VNS2	50311.64	104.68
		VNS3	50370.82	50.99
		GA-FULL	50443.36	115.25
		GA-ONE	49288.89	390.47
		GA-RAND	50499.90*↑⇑	60.91
Miss America, m=6480, d=16,	30, spher., 40 min.	EM	-261971.1	99.6
		CEM	-262265	51.6
		SEM	-261839.6	26.9
		VNS1	-261741.9	10.0
		VNS2	-261737.9	8.7
		VNS3	-261737	1.5
		GA-FULL	-261736*↑⇑	1.6
		GA-ONE	-261893,5	71.7
		GA-RAND	-261763,1	22.3
BIRCH-3 (UCI), m=100000, d=2,	100, spher., 60 min.	EM	-2567483.0	4351,1
		CEM	-2603150.9	5170,3
		SEM	-2728547	3869,7
		VNS (best)	-	-
		GA-FULL	-2553037.9	8014,0
		GA-ONE	-2348371*↑⇑	588278,1
		GA-RAND	-2454123.1	55965.2

Table 2: Comparative results

Tables 1 and 2 show the average results of 30 runs of each algorithm for various datasets. The statistical significance of the difference between the results of the best of new algorithms (marked by "*") in comparison with the EM algorithm was estimated with both Students t-test [34, 35] and Wilcoxon rank-sum test [36, 37]. If the advantage or disadvantage of the best of new algorithms is statistically significant in accordance with Student's t-test, this result is marked by "↑" or "↓", respectively. Advantages and disadvantages in accordance with the Wilcoxon rank-sum test are marked by "⇑" or "⇓", respectively. For both tests, the significance level is 1%.

7. CONCLUSIONS

New genetic evolutionary algorithms and new VNS algorithms for the problems of mixture Gaussian distribution separation allow us to obtain more precise results in comparison with the classical EM-algorithm and its modifications running in the multistart mode. Our new algorithm is a modification of the Greedy Heuristics Method [13, 14, 22], which is efficient for solving many clustering problems including the problem of separating the homogeneous production batches of the electronic devices for spacecraft industry. Though known Variable Neighbourhood Algorithm gives rather precise results in shorter time, new genetic algorithms can be used when we must obtain the most precise results in comparison with other known methods. This paper represents the intermediate stage of research, the set of developed algorithms should serve as a basis for further studies.

Acknowledgment: Results were obtained in the framework of the state task 2.5527.2017/8.9 of the Ministry of Education and Science of the Russian Federation. Authors gratefully acknowledge the anonymous referees for their useful comments and suggestions.

REFERENCES

[1] Orlov, V. I., Stashkov, D. V., Kazakovtsev L. A., Stupina A. A., "Fuzzy clustering of EEE components for space industry", *IOP Conference Series: Materials Science and Engineering*, 155 (2016) Article ID 012026.

[2] Korolev, V. Yu., *EM-algorithm, its modifications and their application to the problem of separation of mixtures of probability distributions*, Theoretical review, IPIRAN, Moscow, 2007.

[3] Bishop, C., *"Pattern Recognition and Machine Learning"*. Springer, 2006.

[4] Newcomb, S., "A generalized theory of the combination of observations so as to obtain the best result", *American Journal of Mathematics*, 8 (4) (1886) 343–366.

[5] Pearson, K., "Contributions to the Mathematical Theory of Evolution", *Philosophical Transactions of the Royal Society of London*, 185 (1894) 71–110.

[6] Everitt, B. and Hand, D. J., *Finite Mixture Distributions, Monographs on Applied Probability and Statistics*, Chapman and Hall, UK, 1981.

[7] McLachlan, G. J., and Basford, K. E., *Mixture models. Inference and applications to clustering*, Marcel Dekker, New York, 1988.

[8] Ayvazyan, S. A., Buhshtaber, V. M., Enyukov, I. S., and Meshalkin, L. D., *Applied Statistics: Classification and Dimension Reduction* [Prikladnaja statistika: klassifikacija i snizhenie razmernosti], Finansy i statistika, Moscow, 1989.

[9] McKendrick, A. G., "Applications of mathematics to medical problems", *Proceedings of the Edinburgh Mathematical Society*, 44 (1926) 98–130.

[10] Healy, M. J. R., and Westmacott, M. H., "Missing Values in Experiments Analyzed on Automatic Computers", *Applied Statistics*, 5 (1956) 203–206.

[11] Shlezinger, M. I., "The Interaction of Learning and Self-Organization in Pattern Recognition", *Kibernetika*, 4 (2) (1968) 81–88.

[12] Dempster, A., Laird, N., and Rubin, D., "Maximum Likelihood Estimation from Incomplete Data" *Journal of the Royal Statistical Society*, Series B, 39 (1977) 1–38.

[13] Kazakovtsev, L. A., and Antamoshkin, A. N., "Greedy Heuristic Method for Location Problems", *Vestnik SibGAU*, 16 (2) (2015) 317–325.

[14] Kazakovtsev, L. A., and Antamoshkin, A. N., "Genetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems", *Informatica*, 3 (38)(2014) 229–240.

[15] Celeux, G., and Govaert, A., *Classification EM Algorithm for Clustering and Two Stochastic Versions*, Rapport de Recherche de lINRIA 1364, Centrede Rocquencourt, 1991.

[16] Celeux, G., and Diebolt, J., "The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem", *Computational Statistics Quarterly*, 2 (1)(1985) 73–82.

[17] Picchini, U., and Samson, A., "Coupling Stochastic EM and Approximate Bayesian Computation for Parameter Inference in State-space Models", *Cornell University Library*, 2017, arXiv:1512.04831v6, DOI:10.1007/s00180-017-0770-y.

[18] Matarazzo, T.J., and Pakzad, S.N., "STRIDE for Structural Identification using Expectation Maximization: Iterative Output-Only Method for Modal Identification", *Journal of Engineering Mechanics*, 142 (2016), DOI:10.1061/(ASCE)EM.1943-7889.0000951.

[19] Zaheer, M., Wick, M., Tristan, J.-B., Smola A., and Steele, G., *Exponential Stochastic Cellular Automata for Massively Parallel Inference*, Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, 2016, 966–975.

[20] Zinovyev, A., "Data Visualization in Political and Social Sciences", *International Encyclopedia of Political Science*, Sage Publications, CA, 2011.

[21] Yin, J., Zhang Y., and Gao L., *Accelerating Expectation-Maximization Algorithms with Frequent Updates*, Proceedings of the IEEE International Conference on Cluster Computing, Beijing, 2012, 275–283.

[22] Kazakovtsev, L. A., Orlov, V. I., Stupina, A. A., and Kazakovtsev, V. L., "New Genetic Algorithm with Greedy Heuristic for Clustering Problems with Unknown Number of Clusters", *Facta Universitatis, series Mathematics and Informatics*, 30(1) (2015) 89–106.

[23] Kaufman, L., and Rousseeuw, P. J., "Clustering by Means of Medoids", in: Y., Dodge (ed.), *Statistical Data Analysis Based on the L1Norm and Related Methods*, North-Holland, 1987, 405–416.

[24] Mladenovic, N., and Hansen, P. "Variable Neighborhood Search", *Comput. Oper. Res.*, 24 (1997) 1097–1100.

[25] Hansen, P., Brimberg, J., Urosevic, D., and Mladenovic N., "Solving Large p-Median Clustering Problems by Primal Dual Variable Neighborhood Search", *Data Mining and Knowledge Discovery*, 19(3) (2009) 351–375.

[26] Hansen, P., "Variable Neighborhood Search", *Search Methodology*, in: E.K. Bruke, and G. Kendall (eds.), Springer, US, (2005) 211–238.

[27] Brimberg, J., Hansen, P., and Mladenovic, N., "Attraction Probabilities in Variable Neighborhood Search", *4OR: A Quarterly Journal of Operations Research*, 8 (2010) 181–194.

[28] Hansen, P., Mladenovic, N., and Moreno Perez, J. A., "Variable Neighborhood Search: Methods and Applications", *4OR: A Quarterly Journal of Operations Research*, 6 (2008) 319–360.

[29] Martins, P., "Goal Clustering: VNS Based Heuristics", *Cornell University Library*, 2017, arXiv:1705.07666.

[30] Stashkov, D. V., "Variable Neighborhood Search Algorithms for Problem of Mixture Distributions Separation", *Sistemy upravleniya i informacionnye tehnologii*, 1(67) (2017) 18–24.

[31] Dua, D., and Karra Taniskidou, E., *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science, 2017.
[32] Franti, P. et al, "Clustering Datasets" [http://cs.uef.fi/sipu/datasets/], 2015.
[33] Nielsen, S. F., "The Stochastic EM Algorithm: Estimation and Asymptotic Results", *Bernoulli*, 6 (2000) 457–489.
[34] Smucker, M. D., Allan, J. and Carterette, B., "A Comparison of Statistical Significance Tests for Information Retrieval", *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM '07)*, ACM, New York, 2007, 623–632.
[35] Park, H. M., "Comparing Group Means: The t-Test and One-way ANOVA Using STATA, SAS, and SPSS", Indiana University, 2009.
[36] Mann, Henry B., Whitney, Donald R., "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other", *Annals of Mathematical Statistics*, 18 (1) (1947) 50–60.
[37] Fay, Michael P., Proschan, Michael A., "WilcoxonMannWhitney or t-Test? On Assumptions for Hypothesis Tests and Multiple Interpretations of Decision Rules", *Statistics Surveys*, 4 (2010) 1–39.