FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 31, No 4 (2016), 907–917

NEW GENETIC ALGORITHM WITH GREEDY HEURISTIC FOR CLUSTERING PROBLEMS WITH UNKNOWN NUMBER OF GROUPS

Lev A. Kazakovtsev, Victor I. Orlov and Vladimir L. Kazakovtsev

Abstract. In this research, we propose new modification of the genetic algorithm with greedy heuristic witch allows to solve series of clustering problems with unknown number of groups (clusters). The applicability and efficiency of new method is experimentally proved. Achieved results are compared with other algorithms which solve each of problems separately. Experiments show that new algorithm is faster than other methods for various problems.

Keywords: Clustering, genetic algorithms, greedy heuristic, optimization.

1. Introduction, problem statement

The most popular clustering problems arise from the location theory and come to p-median problem where attachment of object to a cluster is determined by the attachment of the object to the corresponding center or centroid. If objects are placed in a d-dimensional space with Euclidean or any other metric, we have a p-median problem. In case of squared Euclidean distances, we have the k-means problem. If centers (centroids) are selected from the set of objects only, we have the k-medoid problem [2, 3]. If they are placed on a network, we have a p-median problem on a network [4, 5]. Dimension of the space d is a number of considered characteristics of the objects. Methods of cluster analysis are similar to methods of location theory. In the Weber problems, it is required to calculate coordinates of point to minimize the sum of weighted distances to from other N known points to the closest selected point.

The similar p-median problem in a continuous d-dimensional space is one of generalizations of the Weber problem:

$$\arg \min_{X_1,...,X_p \in R^d} f(X_1, ..., X_p) = \arg\min \sum_{i=1}^{N} w_i \min_{j\in\{\overline{1,p}\}} L(X_j, A_i). \tag{1.1}$$

Received June 30, 2016.; Accepted September 24., 2016.
2010 *Mathematics Subject Classification.* Primary 90B80; Secondary 90B06

Here, $A_i \in R^d, A_i = (a_{i,1}, ..., a_{i,d}), i = 1, ..., N$ are known points (coordinates of objects also called data vectors), $w_i \in R, w_i \geq 0$ are their weight coefficients, $L(\bullet)$ is a metric in a d-dimensional space or any function (measure) of distance. We will call problem with Euclidian metric $|| \bullet ||_2 : R^2 \to R$ "classical p-median problem". Here, $\{X_i | j = \overline{1,p}\}$ is a set of coordinates of new placed objects. If $X_1, ..., X_p \in \{A_i\}$, we have the p-medoid problem. Problem is NP-hard [6] (its complexity grows exponentially). The k-median problem which is the most popular model of clustering problem is a special case of Weber problem. In this research, we proposed and realized new algorithm for clustering problems with unknown number of groups that allows to find rather precise solutions several times faster than known algorithms.

2. Known methods

We can use the following methods for the problems of combinatorial optimization mentioned above:

Genetic Algorithms (GA) [10]. This heuristic is used to generate useful solutions to optimtzation problems. Every solution is an "individual" in a "population" which is coded in some specific way. In classical GA, primary population is generated randomly, every indivdual is coded as an L-bit string ("chromosome"), where L is the length of a string (number of bits), length of coding is the similar for every solution. The main process in this algorithm is the crossingover procedure that recombines old solutions and creates the new one. [10] First genetic algorithm for solving the p-median problem on a network was proposed in [11]. In [4, 12], authors propose algorithms which give sufficiently accurate results but take a long time. In [13], authors propose an algorithm that recombines solutions with specific "greedy" heuristic.

"Greedy" heuristic algorithms [1]. Mostly, they are used in combination with other algorithms. The advantage of the greedy methods is that its computational complexity grows linearly with N. This heuristic cannot guarantee finding the optimal solution [8], however, we can make theoretical lower bounds of the solution [1]. Greedy heuristic combined with the GA creates acceptable results for location problems. Efficiency of this algorithms for k-medoid problem is statistically proved [13].

GA with greedy heuristic was created for p-median problem (and similar problems). In [14], it was adapted for pseudo-Boolean optimization problems such as a knapsack problem. Difference with classical GA is in the crossingover procedure. In this algorithm, the "chromosome" status of "child individual" depends on its status of "parent solution". However, in case when an element of the parent "chromosome" equals to 1, the element of the child "chromosome" equals to 1 (thus, this algorithm joins two parent solutions). However, in "classical" GA, if parent "chromosomes" are different, the value in the child "chromosome" is randomly selected (randomly choosing 1 or 0). Solutions created by this iteration do not satisfy to the problem constraints (number of centers must be equal to p, thus, the "chromosome" must contain p elements equal to 1).

Multiple location problems like p-median, discrete or continuous, are the global search problems. For many metrics and other distance measures, local search algorithms (they improve created result) have already been designed. Usually, they contain the ALA procedure (Alternating Location-Allocation) that finds a local minimum of the objective function (1.1) starting from known initial points (centers) $X_1, ..., X_p$ which are chosen from data vectors.

Algorithm 2.1. ALA method [6]

Require: Set $V = (A_1, ..., A_N)$ of N data vectors in d-dimensional space, $A_1 = (a_{1,1}, ..., a_{1,d}), ..., A_N = (a_{N,1}, ..., a_{N,d})$, initial solution: a set of centers or centroid of p clusters $X_1 = (x_{1,1}, ..., a_{1,d}), ..., X_p = (x_{p,1}, ..., x_{p,d})$.

Step 1: For each data vector, find the closest centroid:

$C_i = \arg\min_{j=\overline{1,p}}, L(A_i, X_j) \forall i = \overline{1, N}$.

Thus, p clusters are formed.

Step 2: For each cluster $\mathcal{C}_j^{clust} = \{i \in \{\overline{1,N}\} | C_i = j\}$, recalculate its center or centroid X_j. In the case of Euclidean (l_2) metric, Weiszfeld procedure or its advanced modification can be used. In the case of squared Euclidean (l_2^2) metric, each of d coordinates are calculated as the mean value of this coordinate values among the cluster.

Step 3: If any center or centroid was altered at Step 2 then go to Step 1.

Step 4: Otherwise, STOP. $X_1, ..., X_p$ are local minima.

Procedure can be used on networks and on continuous space. The convergence to a local minimum is proved [7]. We can also use some heuristics for solving location or clustering problems. Next one is one of the most efficient.

Algorithm 2.2. GA with greedy heuristic for p-median problem [4]

Step 1: Initialize a population of N_{pop} individuals. Every individual is a set of p_{max} centers (we denominate it X where X_i is $i^t h$ element of set). Assign $F_{new,j} = +\infty$ for $j = \overline{1, N_{pop}}$. Initialize array of best values of objective function $F_k^* = +\infty$ and best solutions $X_k^* = \{\}$ for $k = \overline{2, p_{max}}$.

Step 2: Choose randomly $j_1, j_2 \in [1, N], j_1 \neq j_2$

Step 3: $X_{new} = X_{j_1} \cup X_{j_2}$

Step 4: While $|X_{new}| > p_{max}$:

Step 5: Choose

$j = \arg\min j \in \chi_{j3} F(\chi_{j3} \setminus \{j\}) = \arg\min_{j \in \chi_{j3}} \sum_{i=1}^{N} \min_{j' \in (\chi_{j3} \setminus j)} w_i L(i,j)$

Step 6: $\chi_{j3} = \chi_{j3} \setminus \{j\}$. Improve solution χ_{j3} by the local search.

Step 7: Next iteration of cycle 4.

Step 8: Check the stop conditions, go to 2.

This algorithm has some interesting properties.

Authors determine the size of initial population as

$$N_{POP} = \lceil NC\frac{\binom{N}{p}}{100\lceil N/p\rceil}\rceil\lceil N/p\rceil.$$

However, algorithm gives good result at much lower population (e.g. $N_{POP} = 20$). Herewith, as opposite to original GA with greedy heuristic, which initial population is initialized with a deterministic procedure, initial population is needed to be initialized randomly.

The Algorithm is combined with local search on Steps 5 and 6 which allows to create an acceptable solutions in less time but often accuracy is lower.

Computational complexity of one iteration needs $\lceil 1,5p^2\rceil$ computations of the objective function and it is the problem of local search algorithms which can be eliminated by some modifications:

3. New method

In this research, we propose a new method for solving clustering problems. Algorithm is a development of genetic algorithm with greedy heuristic [15] and its main difference is that it can solve such problems when we do not know how many clusters we have and it is required to solve a series of problems with various values of $p = \overline{2, p_{max}}$.

An example of this problem is the problem of the additional screening test of electronic chips for spaceships [16] and other special purposes. It is essential that all components are guaranteed to be homogeneous and produced as a single production batch. We need to solve problem for all possible values of p at once.

Usual algorithms, for example k-medoid or very efficient algorithm with recombination of sets of centers with fixed power (fixed length) that was improved by Sheng and Liu [3] require value of p to be indicated. Algorithms like X-means choose the best value of p by certain criteria which adequacy is not obvious.

We offer some easy modification of GA with greedy heuristic: do not stop the elimination of clusters after achievement p clusters and with every iteration fix the objective function value.

Algorithm 3.1. GA with greedy heuristic for solving series of problems with $p = \overline{2, p_{max}}$

```
Step 1: Initialize a population of $N_{pop}$ individuals. Every individual is a set of $p_{max}$ centers (we denominate it $X$ where $X_i$ is $i^{th}$ element of set). Assign $F_{new,j} = +\infty$ for $j = \overline{1, N_{pop}}$. Initialize array of best values of objective function $F_k^* = +\infty$ and best solutions $X_k^* = \{\}$ for $k = \overline{2, p_{max}}$.
Step 2: Choose randomly $j_1, j_2\ in [1, N], j_1 \neq j_2$
Step 3: $X_{new} = X_{j_1} \cup X_{j_2}$
Step 4: While $|X_{new}| > p_{max}$ :
Step 4.1: Choose object index $j$ which gives the smallest increase of the objective function after its deleting: $j = \arg\min_{j \in \chi_{new}} F(\chi_{new} \setminus \{j\})$
Step 4.2: $X_{new} = X_{new} \setminus \{j\}$. Continue the cycle 4.
Step 5: Assign $F_{new} = 0$ Here, we introduce the new utility function, a sum of objective function values for $p = \overline{2, p_{max}}$ Assign $X^* = X_{new}$
Step 6: While $|X_{new} > 2|$,
Step 6.1: Assign $F_{new} = F_{new} + f(X_{new})$; $k = |X_{new}|$; $F_k = f(X_{new})$; if $F_k < F_k^*$ than $F_k^* = F_k$
Step 6.2: Perform steps 4.1 and 4.2 for $X_{new}$
Step 6.3: Continue cycle 6.
Step 7: Choose $j_3$ by tournament selection by value of $F_{new,j}$. Assign $F_{j_3} = F_{new}$; $X_{j_3} = X^*$, $F_{new,j_3} = F_{new}$
Step 8: Check the stop conditions, go to Step 2.
```

We sum the objective function values for every value of p and now we have a new utility function. Investigation of applicability of this algorithm to the problems of cluster analysis and comparison with other algorithms presented below. First results are submitted in Table 3.1-3.2 and Fig. 3.1-3.2.

Results in Fig. 3.1-3.2 show that our algorithm is not so accurate as GA by Sheng and Liu for some problems (look "Chess" data base). But for many other problems, our algorithm gives good result solving a series of large-scale problems at once. In addition, it can solve very large problem (e.g. KDD Cup Dataset).

For continuous p-median problems (Table 3.1-3.2) in conjunction with various methods of local search, results was received for problems with a comparatively big number of clusters. Having reduced this number of clusters, the effectiveness of the algorithm is also reduced. Thus, we suppose that the algorithm should be stopped earlier than at $p = 2$, approximately at $p = p_{max}/3$. Further solutions can be received by other methods or by new start of our method with less value of p_{max}.

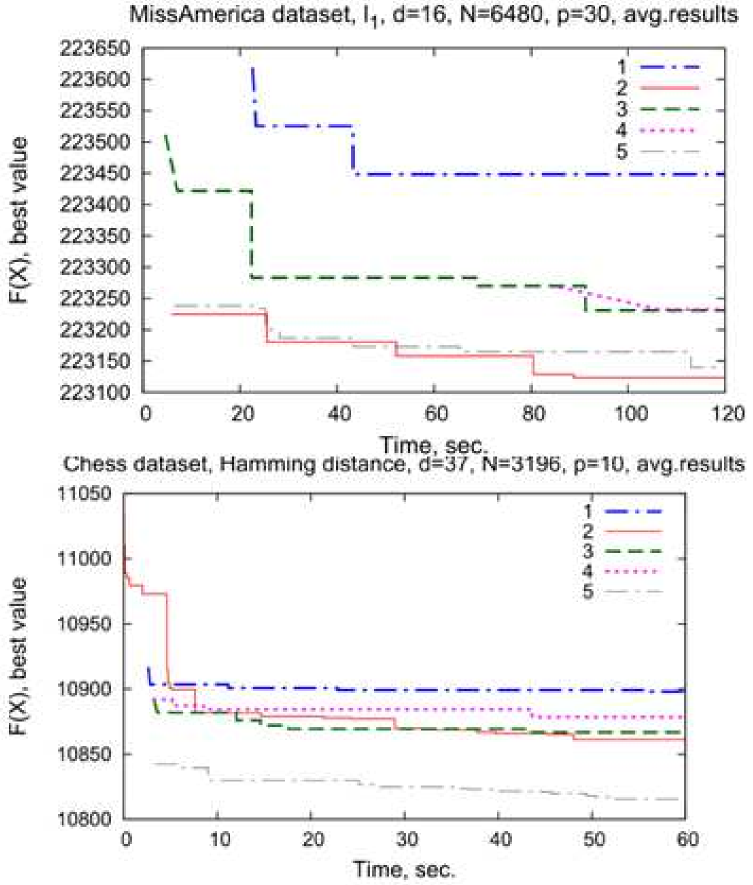

Fig. 3.1: Dynamics of change of the value of the objective function, averaged over 10 starts of algorithm for k-medoid problems. Lines description: 1: Multistart of local search in SWAP neighbourhood, 2: GA with greedy heuristic combined with ALA procedure, 3: GA with greedy heuristic for solving series of problems combined with SWAP neighbourhood search, 4: GA with greedy heuristic for solving one problem in combination with local search in SWAP neighbourhood, 5: GA by Sheng and Liu in combination with local search in SWAP neighbourhood, 6: multistart of PAM procedure.

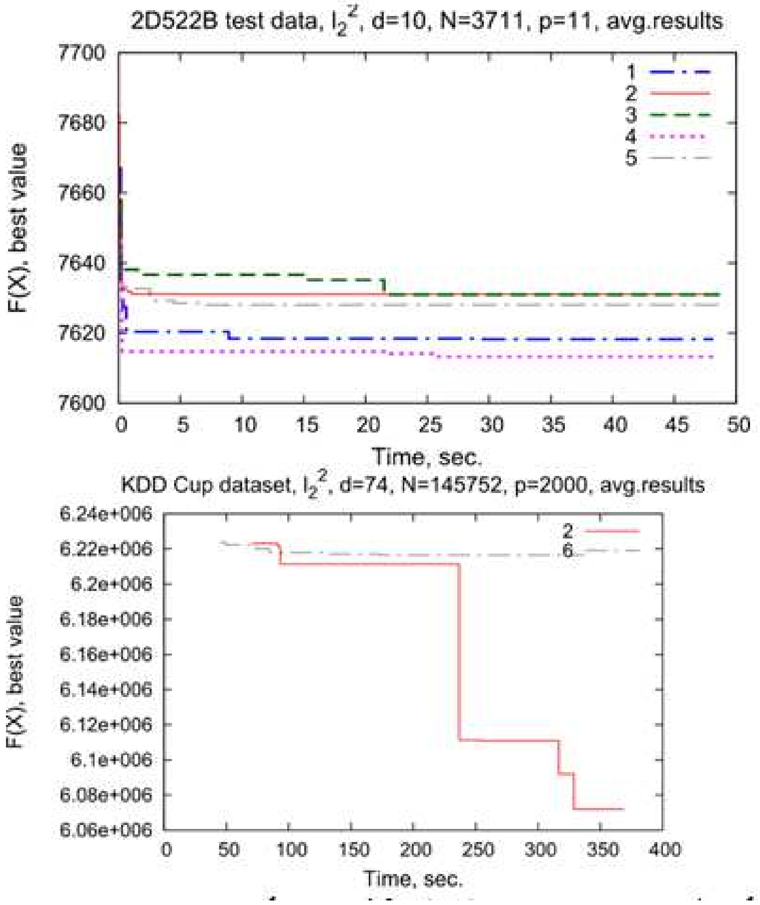

FIG. 3.2: Dynamics of change of the value of the objective function, averaged over 10 starts of algorithm for k-medoid problems. Lines description: 1: Multistart of local search in SWAP neighbourhood, 2: GA with greedy heuristic combined with ALA procedure, 3: GA with greedy heuristic for solving series of problems combined with SWAP neighbourhood search, 4: GA with greedy heuristic for solving one problem in combination with local search in SWAP neighbourhood, 5: GA by Sheng and Liu in combination with local search in SWAP neighbourhood, 6: multistart of PAM-procedure.

Table 3.1: Comparative results of various algorithms, part 1

Data set and its parameters	Clusters quantity	Algo-rithm	Time, sec.	Avg. result	Avg. deviation
1	p =14	1	15	150,124869801	0,384203928
		2	15	150,533299444	0,598587789
		3	15	149,954679652	0,172789313
		4	15	151,280175427	0,982922979
		5	15	149,78736565*	0,03157532*
		6	15	151,082443691	0,654212395
	p =10	1	15	198,375350991	0,018643710
		2	15	198,426881563	0,044039446
		3	15	198,377650812	0,024878118
		4	15	198,450402498	0,032311263
		5	15	198,359747028	2×10^{-14}*
		6	15	198,35421865*	0,0070903
	p =6	1	15	362,70701636*	0*
		2	15	362,70701636*	0*
		3	15	362,70701636*	0*
		4	15	362,704156850	0,000344112
		5	15	362,704051312	0*
		6	15	362,704051312	0*
2	p =10	1	15	359,680203232	3,964320582
		2	15	359,545287242	0,208756158
		3	15	359,545250068	2,526439494
		4	15	361,435624000	0,208770779
		5	15	359,410460803	0,177992934
		6	15	359,41036391*	0*
	p =4	1	15	596,825210394	0,000000442
		2	15	596,825217410	0,000004148
		3	15	596,82520843*	0,000000388
		4	15	596,825208927	0,000000574
		5	15	596,825283111	0*
		6	15	596,825283111	0*

Table 3.2: Comparative results of various algorithms, part 2

Data set and its parameters	Clusters quantity	Algo-rithm	Time, sec.	Avg. result	Avg. deviation
3	p =100	1	30	$3,7513245 \times 10^{13}$	116786778766
		2	3000	$3,7711179 \times 10^{13}$	158613580914
		5	30	$3,740432 \times 10^{13}$*	21699776156*
		6	30	-	-
	p =50	3	30	$9,0099578 \times 10^{13}$	9545892119
		4	30	-	
		5	30	$8,902789 \times 10^{13}$*	0*
		6	30	-	
	p =20	3	30	$3,303278 \times 10^{14}$*	0*
		4	30	-	-
		5	30	$3,3049972 \times 10^{14}$	0*
		6	30	-	-

Note: (Problems) 1: Screening test of 1526TL1 chip, N=1234, d=120. (in 5 and 6 - $p \in \{2..20\}$); 2: UCI Mopsi Joensuu, N=6014, d=2. (in 5 and 6 - $p \in \{2..20\}$); 3: BIRCH-3, N=100000, d=2. (in 5 and 6 - $p \in \{2..110\}$) (Algorithms) 1: Multistart of procedure k-means (ALA-procedure), 2: multistart of procedure j-means, 3: GA with recombination of subsets with fixed length (Sheng, Liu) combined with ALA-procedure as a local search algorithm, 4: same GA combined with j-means procedure as local search algorithm, 5, 6: GA with greedy heuristic with a real alphabet for solving series of problems in combination with ALA procedure and procedure j-means (here, time is indicated for every number of p). Symbol ”*” marks the best result.

4. Conclusion

In addition to adaptation of greedy heuristics to other class of problems, we create its modification that allows to solve problems when we do not know how much clusters we have. For example, a problem of grouping of electronic chips into homogeneous production batches while testing them can be solves for every possible number of production batches at once without loose of accuracy.

REFERENCES

1. G.N. Diubin and A.A. Korbut: *Greedy algorithms for the knapsack problem: behavior in the mean.* Sibirskii jurnal industrialnoi matematiki **2, issue 2** (1999), 68-93.
2. L.A. Kazakovtsev: *Evolutionary algorithm for k-medoid problem.* Sistemi upravlenia I informatsionnie tehnologii. **1** (2015), 95-99.
3. W. Sheng and X. Liu: *A Genetic K-Medoids Clustering Algorithm.* Journal of Heuristics. **12, issue 6** (2004), 447-466.
4. O. Alp, E. Erkut and Z. Drezner: *An Efficient Genetic Algorithm for the p-Median Problem.* Annals of Operations Research. **122, issue 1-4** (2003), 21-42.
5. A.N. Antamoshkin and L.A. Kazakovtsev: *Random Search Algorithm for the p-Median Problem.* Informatica. **37** (2013), 267-278.
6. L. Cooper: *Location-allocation problem.* Oper. Res. **11** (1963), 331-343.
7. D. Arthur, B. Manthey and H. Roglin: *k-Means Has Polynomial Smoothed Complexity.* Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS '09), Washington:IEEE Computer Society (2009), 405-414, DOI: 10.1109/FOCS.2009.14.
8. L.A. Rastrigin: *This random, random, random world.* Molodaia gvardia, Moscow, 1974.
9. A.N. Antamoshkin: *Optimizing of functional with Boolean variables.* Publishing house of Tomsk university, Tomsk, 1987
10. G.K. Voronovskii, K.V. Mahotilo, S.N. Petrashev and S.A. Sergeev: *Genetic algorithms, artificial neural networks and virtual reality problems.* Osnova, Kharkov, 1997
11. C.M. Hosage and M.F. Goodchild: *Discrete Space Location-Allocation Solutions from Genetic Algorithms.* Annals of Operations Research. **6** (1986), 35-46.
12. Z. Drezner, H. Hamacher and other: *Facility Location: Applications and Theory.* Springer, (2002).
13. L.A. Kazakovtsev, V.I. Orlov, A.A. Stupina and V.L. Kazakovtsev: *Modified Genetic Algorithm with Greedy Heuristic for Continuous and Discrete p-Median Problems.* Facta Universitatis Series Mathematics and Informatics. **30, issue 1** (2015), 89-106.
14. V.L Kazakovtsev and L.A. Kazakovtsev: *About methods of solving of big combinatorial problems.* Collection of research papers of II international scientific-practical conference of students "Osennii shkolnii marafon". (2014), 72-76.

15. L.A. Kazakovtsev and A.N. Antamoshkin: *Method of greedy heuristics for location problems.* Vestnik SibGAU. **16, issue 2** (2015), 317-325,

16. N.V. Kopliarova, V.I. Orlov, N.A. Sergeeva and V.V. Fedosov: *About nonparametric models in electronic chips diagnostic problems.* Zavodskaia laboratoria. Diagnostika materialov. **80, issue 7** (2014), 73-77.

Lev A. Kazakovtsev
Siberian State Aerospace University
Department of Information Technologies
prosp.Krasnoyarskiy Rabochiy, 31
660014 Krasnoyarsk, Russian Federation
levk@bk.ru

Victor I. Orlov
TTC NPO PM
ul. Molodezhnaya, 20
662970 Zheleznogorsk, Krasnoyarskii Krai, Russian Federation
ttc@krasmail.ru

Vladimir L. Kazakovtsev
Siberian Federal University
School of Physics and Mathematics
prosp. Svobodnii, 79
660041 Krasnoyarsk, Krasnoyarskii Krai, Russian Federation
vokz@bk.ru