Combinations of the Greedy Heuristic Method for Clustering Problems and Local Search Algorithms

Lev Kazakovtsev[1,2,*], Alexander Antamoshkin[1,2]

[1]Siberian State Aerospace University Named after Academician M.F.Reshetnev, Krasnoyarsk, Russian Federation
[2]Siberian Federal University, Krasnoyarsk, Russian Federation
levk@bk.ru, oleslav@mail.ru

Abstract. In this paper, we investigate application of various options of algorithms with greedy agglomerative heuristic procedure for object clustering problems in continuous space in combination with various local search methods. We propose new modifications of the greedy agglomerative heuristic algorithms with local search in SWAP neighborhood for the p-medoid problems and j-means procedure for continuous clustering problems (p-median and k-means). New modifications of algorithms were applied to clustering problems in both continuous and discrete settings. Computational results with classical data sets and real data show the comparative efficiency of new algorithms for middle-size problems only.

Keywords: p-median · k-means · p-medoids · Genetic algorithm · Heuristic optimization · Clustering

1 Introduction

Problems of automatic groupping of objects in a continuous space with defined distance measure function between two points are the most widely applied clustering models. The *k*-means problem is most popular clustering model in which it is required to split *N* objects into *k* groups so that the sum of squares of distances from objects to the closest center of a group reaches its minimum. Centers (sometimes called by centroids) are unknown points in the same space. Other popular clustering model is the *p*-median problem [9] which has similar setting but instead of the sum of distance squares, sum of distances has to be minimized. Thus, in the *k*-means problem, a measure of distance is the squared Euclidean distance. The similarity of the continuous *p*-median problem and *k*-means problem was emphasized by many researchers [25, 12, 16, 13].

There exists a special class of discrete optimization problems operating concepts of the continuous problems called *p*-medoid problem [19] or discrete *p*-median problem [37]. Such clustering problems can be considered as location problems [34, 28] since the main parameters of such continuous and discrete problems are coordinates of objects and distance between them [10, 9]. Such problems arise in statistics (for

In: A. Kononov et al. (eds.): DOOR 2016, Vladivostok, Russia, published at http://ceur-ws.org

example, problems of estimation), statistical data processing, signal or image processing and other engineering applications [20].

The formulation of the p-median problem [12, 7] is as follows:

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

Here, $\{A_i | i = \overline{1,N}\}$ is a set of known points (data vectors), $\{X_j | j = \overline{1,p}\}$ are unknown points (centers, centroids), w_i are weight coefficients (usually equal to 1), $L(\cdot)$ is the distance function in a d-dimensional space [14, 9]. In most popular cases, $L(\cdot)$ is squared Euclidean distance, Euclidean or Manhattan metric.

The center of each cluster (group) is the solution of the 1-median (Weber) problem for this cluster. If the distance measure is squared Euclidean distance (l_2^2) then the solution is [12]:

$$x_j = \sum_{i=1}^{N} w_i a_{i,j} / \sum_{i=1}^{N} w_i. \tag{2}$$

Here, $X = (x_1,\dots,x_d)$, $A_i = (a_{i,1},\dots,a_{i,d}) \forall i = \overline{1,N}$.

Such p-median, k-means and k-medoids problems are problems of general optimization: objective function is non-convex [8]. Moreover, they are NP-hard [11, 2, 33, 40, 41, 15]. The most popular ALA procedure for such problems is based on the Lloyd algorithm [30] also known as standard k-means procedure [32]. Nevertheless, many authors offered faster methods based on this standard procedure [46, 1] for data sets and data streams. The ALA and similar procedures are algorithms sequentially improving the known solution. They are not true local search algorithms in strict sense since the new decision is searched not necessarily in a ε-neighborhood of the known solution.

The modern literature offers many heuristic methods [36] for seeding of the initial solutions (sets of centers) for the ALA procedure which are various evolutionary methods and random search methods such as k-means++ procedure [4].

Popular idea is use of the genetic algorithms (GA) and for improving of results of local search [18, 27, 39]. In case of GAs, authors use various methods of coding of solutions which form a "population" of the evolutionary algorithm. Hosage and Goodchild [17] offered the first genetic algorithm for the *p*-median problem on a network. Rather precise but very slow algorithm based on special greedy agglomerative heuristics was offered by Bozkaya, Zhang and Erkut in 2002 [46].

Rather precise and fast algorithm with the greedy agglomerative heuristic recombination procedure for the *p*-median problem on a network was offered by Alp, Erkut and Drezner in 2002 [3]. This algorithm was adapted for the continuous problems by Neema et al. in 2011 [39]. At each iteration, this algorithm generates initial solutions for the local search procedure. Thus, this approach becomes extremely slow with growth of number of clusters p. Other method of recombination is offered by Sheng and Liu (2006) [43]. These algorithms work quicker, however, they are less precise. Also Lim and Xiu in 2003 offered the genetic algorithm based on recombination of subsets of centers of the fixed length [29].

The genetic algorithm with greedy heuristics [3] does not provide use of a mutation procedure which is common for GAs [38].

In [21], authors propose the GA with greedy agglomerative heuristic using floating point alphabet. Most GAs for p-median problems [39] use the integer alphabet for coding of initial solutions of the ALA procedure by numbers of corresponding data vectors. In [21], authors encode the interim solutions in a form of sets of points (centers) in the d-dimensional space which are changed by the ALA procedure. Such combination of greedy agglomerative heuristics and ALA procedure allows algorithm to receive more precise results.

In this paper, we propose further modification of the genetic algorithm with greedy heuristic which includes combination of the greedy agglomerative heuristic procedures with local search in wider neighborhoods such as SWAP neighborhood.

2 Known algorithms

The ALA procedure includes two simple steps:

Algorithm 1. ALA procedure.

Required: data vectors $A_1...A_N$, k initial cluster centers $X_1...X_k$.

1. For each center X_i, determine its cluster C_i as a subset of the data vectors for which this center X_i is the closest one. $C_i = \arg\min_{j=\overline{1,p}} L(A_i, X_j) \forall i = \overline{1,N}$.

2. For each cluster $C_j^{clust} = \{i \in \{\overline{1,N}\} | C_i = j\}$, recalculate its center X_i (i.e., solve the Weber problem).

3. Repeat Step 1 unless Steps 1, 2 made no change in any cluster.

Many papers propose approaches to improve the speed of the ALA algorithm [1] such as random sampling [35] etc.

The GA with greedy heuristic [3, 23] with modifications [39, 21, 24] can be described as follows.

Algorithm 2. GA with greedy heuristic for p-median and p-medoid problems.

Required: Population size N_{POP}.

1. Generate (randomly with equal probabilities or using the k-means++ procedure) N_{POP} initial solutions $\chi_1,..\chi_{N_{POP}} \subset \{\overline{1,N}\}$, $|\chi_i| = p \forall i = \overline{1,N_{POP}}$ which are sets of data vectors used as initial solutions for the ALA procedure. For each of them, run the ALA procedure and estimate the values $F_{fitness}(\chi)$ of the objective function (1) for the results of the ALA procedure, save these values to variables $f_1,\ldots,f_{N_{POP}}$.

2. If the stop conditions are reached then STOP. The solution if the set of the initial centers χ_{i^*} corresponding the minimal value of f_i. For estimating the final solution, run the ALA procedure for χ_{i^*} .

3. Choose randomly two indexes $k_1, k_2 \in \{\overline{1,N}\}, k_1 \neq k_2$.

4. Form an interim solution $\chi_c = \chi_{k_1} \cup \chi_{k_2}$.

5. If $|\chi_c|>p$ then go to Step 7:

6. Perform the greedy agglomerative heuristic procedure (Algorithm 3) for χ_c with elimination intensity parameter σ.

7. If $\exists i\in\{\overline{1,N_{POP}}\}:\chi_i=\chi_c$ then go to Step 2.

8. Choose an index $k_3\in\{\overline{1,N_{POP}}\}$. Authors [21] use a simple tournament selection procedure: choose randomly $k_{4,}k_5\in\{\overline{1,N_{POP}}\}$, if $f_{k_4}>f_{k_5}$ then $k_3=k_4$, otherwise $k_3=k_5$.

9. Replace χ_{k_3} and corresponding objective function value: $\chi_{k_3}=\chi_c$, $f_{k_3}=F_{fitness}(\chi_c)$. Go to Step 2.

The greedy agglomerative heuristic is realized as follows:

Algorithm 3. Greedy agglomerative heuristic.

Required: initial solution χ_C , elimination parameter σ.

1.For each $j\in\chi_c$ do:

1.1. Form a set $\chi^-=\chi_c\setminus\{j\}$. For each data vector $A_i\in\{A_1,\ldots,A_N\}$, choose the closest center $C_i^-=arg \underset{j=1,|\chi^-|}{\underline{min}} L(A_i,X_j)$. Form $|\chi^-|$ sets of data vectors for each of which the center is its closest center: $C_j^{clust-}=\left\{i\in\left\{\overline{1,N}\right\}|\,C_i^-=j\right\}$; for each cluster $C_j^{clust-}, j=\overline{1,|\chi_c|}$, calculate its center X_j^-, and then calculate $f_j^-=\Sigma_{i=1}^N w_i \min_{k\in\chi^-} L(X_k^-,A_i)$.

1.2. Next iteration 1.

2. Sort the set of pairs (*j*, $f_j^-=\Sigma_{i=1}^N w_i \min_{k\in\chi^-} L(X_k^-,A_i)$.) in ascending order of f_j^- , form a set E_{elim} of the first N^{elim} indexes of data vectors from this arranged set. From this set E_{elim}, eliminate indexes *i* such that $\exists j\in E_{e\lim}:\|A_i-A_j\|<L_{\min},\quad j<i$.

Here, N_{elim}=[σ (| $\chi_{\rm C}$ |-p)]+1. The value of parameter L_{min} is equal to 0.1-0.5 of the average distance between two data vectors. We use $L_{\min}=0{,}2\,\mu(L(A_i,A_j))$. Parameter σ regulates the process of eliminating of the elements from the interim solution. Value 0 means sequential deleting of elements (centers, centroids, medoids) one by one. The default value 0.2 makes the algorithm work faster which is important if value of *p* is comparatively large (*p*>10).

3. Eliminate E_{elim} from χ_c : $\chi_c=\chi_c\setminus E_{e\,lim}$. and return.

Value of the objective function is calculated depending as follows:

Algorithm 4. Calculating the objective function value $F_{fitness}(\chi)$.

Required: initial solution $\chi = \{X_1,\dots,X_p\}$.

1. Run the ALA procedure or the PAM procedure with the initial solution χ to gen new set of centers $\{X_1,\dots,X_p\}$.

2. Return $F_{fitness}(\chi) = \sum_{i=1}^{N} w_i \min_{j\in\{1,p\}} L(X_j, A_i)$.

Step 4 of Algorithm 2 generates an interim solution set with cardinality up to $2p$ from which we sequentially eliminate elements (Step 6) until we reach cardinality equal to p. At each iteration, the value $F_{fintess}$ is calculated up to $2p$ times. Thus, the local search procedure is started up to p^2 in each iteration of the greedy heuristic recombination. In the case of the k-medoid problem, the computational complexity increases. In this case, we must calculate

$$x_{j,k} = \min_{i\in C_j^{clust}} \sum_{j\in C_j^{clust}} \sum_{k=1}^{d} \sqrt{(a_i - a_j)^2}.$$

Instead of the ALA procedure, we can use faster PAM procedure. Nevertheless, its complexity also depends on p and N.

3 Combination with alternative local search algorithms

The structure of Algorithm 4 can be described as three nested loop.

The first of them performs iterations of iteration of strategy of global search (for the GA, this is execution of evolutionary operators of random selection and starting the crossingover procedure, however, other metaheuristics can be also applied [22]). The second nested loop performs the iterations greedy heuristic procedure until the feasible solution is reached. The third nested loop within this procedure provides estimation of consequences of eliminating each of elements of the intermediate decision.

The steps realizing greedy agglomerative heuristics are launched in combination with one of the existing algorithms of local search. Thus, for the continuous problems, it can be the ALA procedure or other two-step procedures of the alternating location and allocation, for example, the j-means or h-means procdure. Search in different types of neighborhoods can be applied to the discrete clustering problems. The PAM procedure (Partition around Medoid, i.e. neighborhood of the given quantity of the closest data vectors), ALA procedure (wider neighborhood: all data vectors of a cluster), or search in wider SWAP or K-SWAP-neighborhoods can be applied to the p-medoid problem.

For the p-median, p-medoids and k-means problems, we can use the ALA procedure as a local search procedure [44, 31]. The PAM procedure is a search algorithm in neighborhood of each of medoids consisting of the solutions received by changeover of a medoid by one of p_{PAM} of the closest data vectors. In our research, we used p_{PAM} =3. However, experiments did not reveal any universal advantage of each of methods:

PAM allows to have a good result quicker case of a small number of large clusters, ALA procedure is more efficient in the case of small clusters.

Meanwhile, there are many other efficient local search methods the discrete clustering problems [45, 42, 26]. Long ago it was noted [42] that very precise solutions can be obtained with the algorithms based on search in a SWAP neighborhood formed by changeover of a medoid by any data vector. In many cases, search in this neighborhood allows to receive exact solutions [26]. Computing complexity of one iteration of such procedure lies within O(pN^2). Traditionally, this procedure is applied to rather small data sets (N <5000). Larger neighborhoods such as K-SWAP (changeover of K medoids with other data vectors), certainly, are capable to increase the expected accuracy of the result received after the single start of such procedure, but time expenses grow so fast that even in the 2-SWAP neighborhood, search is possible for very small data sets only. In the case when K=p, search in this neighborhood degenerates into the full search and gives the exact solution.

The greedy genetic algorithms mentioned in this paper (including Algorithm 2) use the ALA procedure (as it was mentioned above, in the case of the p-medoid problem, PAM procedure is also possible). In this research, we add search in SWAP neighborhood to this procedure . Algorithm 5 can be used instead of the ALA procedure option in Algorithms 2 and 4.

Algorithm 5. Local search combination for the greedy heuristic algorithm.

Required: initial solution χ which is a set of centers, centroids or medoids.

Step 1. Run the ALA procedure (or the PAM procedure) from the initial solution χ. Store the new value of χ.

Step 2. If Step 1 did not improve the solution then STOP and return χ .

Step 3. Form array I of numbers $\{\overline{1,p}\}$, shuffle this array randomly.

Step 4. For each $i' \in I$ do:

Step 4.1. Store i=$I_{i'}$.Store $f^{'} = +\infty$.

Step 4.2. Store $j^{'} = \arg\min_{j\in\{\overline{1,N}\}} F((\chi \setminus\{\chi_i\} \cup \{A_j\}))$. Here, χ_i is the ith center or centroid or medoid in the solution, A_j is the jth data vector.

Step 4.3. If $F((\chi \setminus\{\chi_i\} \cup \{A_j\}))$<$F(\chi$)), then store $\chi = \chi \setminus\{\chi_i\} \cup \{A_j\}$ and go to Step 1.

Steps of this algorithm combine search in SWAP neighborhood (Steps 4-4.3) with the ALA procedure (Step 1). While the PAM procedure is a simple local search algorithm, the ALA procedure is not a true local search algorithm. However, it improves an existing solution step by step similarly to the local search algorithm.

For the continuous problems (k-means, p-median) there exists the j-means procedure [16] similar to search in SWAP neighborhood which scope is also restricted to comparatively small problems. This procedure is reduced to changeover of centers/centroids by one of data vectors with the subsequent continuation of search by standard ALA procedure. It is easy to note that this principle of search is realized by Algorithm 5. Thus, Algorithm 5 realizes alternation of ALA or PAM procedure with search in SWAP neighborhood for the discrete problems and the j-means procedure for the continuous ones.

We will apply Algorithm 5 instead of Algorithm 4 for solving discrete and continuous problems as a part of genetic algorithms with greedy heuristic (Algorithm 2). For testing purposes, we use the real data and the generated data sets collected by department of processing of the speech and images of computing School of University of Eastern Finland and a repository of machine training of UCI, and also data of tests of EEE components for spaceships [20].

4 Computational experiments

In our experiments, we used the Depo X8Sti computer (6-core CPU Xeon X5650 2.67 GHz, 12Gb RAM). We launched each algorithm with each data set 30 times.

In Fig. 1, 2 and Table 1, some results of computing experiments are provided.
With number of vectors of data N>10000, the attempts to apply Algorithm 5 as a part of the genetic algorithm with greedy heuristics were unsuccessful since the single start of Algorithm 5 required time exceeding all time limit of for the GA.

Dynamics of change of the best known value of the objective function shows that for small discrete problems (N <1000), application of Algorithm 5 has advantage in comparison with Algorithm 4. Moreover, for such problems, application of Algorithm 5 even as a separate algorithm, without use of the genetic algorithm has advantage in comparison with the genetic algorithm with greedy heuristics in a combination with Algorithm 4.

Middle-size problems (*N*<10000) show other situation. In certain cases, the genetic algorithm with greedy heuristics even without application of search in SWAP neighborhood shows the best results. In other cases, application of search in SWAP neighborhood is repaid but application of this type of local search as a part of the genetic algorithm leads to further improving of result. Moreover, in certain cases, simpler recombination procedures of the genetic algorithm such as the genetic algorithm with a recombination of fixed length subsets [43] yield the best results in comparison with GA with greedy heuristics. This is most evident for problems with Boolean and categorical data. Nevertheless, the obtained results are comparable by accuracy and use of the GA with greedy heuristics can be competitive for problems with 1000<*N*<10000.

For the continuous problems, (except the smallest ones with *N*<1000), application of algorithms with greedy heuristics in a combination with Algorithm 5 is quite justified in the majority of practical cases.

5 Conclusions

Application of search in SWAP neighborhood (and probably wider neighborhoods) can be competitive in the case of the small problems, but is not competitive for large-scale problems (*N*>10000). Experiments show that this effect does not depend on the heuristic used for constructing the initial population. As a rule, obtaining an acceptable result is decelerated in case of using SWAP procedure. The exception is some problems with Jacquard metric and Hamming metric. Depending on problem parameters (*N*, *p*, *d*, metric) and time limit and accuracy requirements, various local search procedures are more or less efficient. The most important factor in the case of the large-scale problems is time consumption for a single run of the SWAP local search procedure.

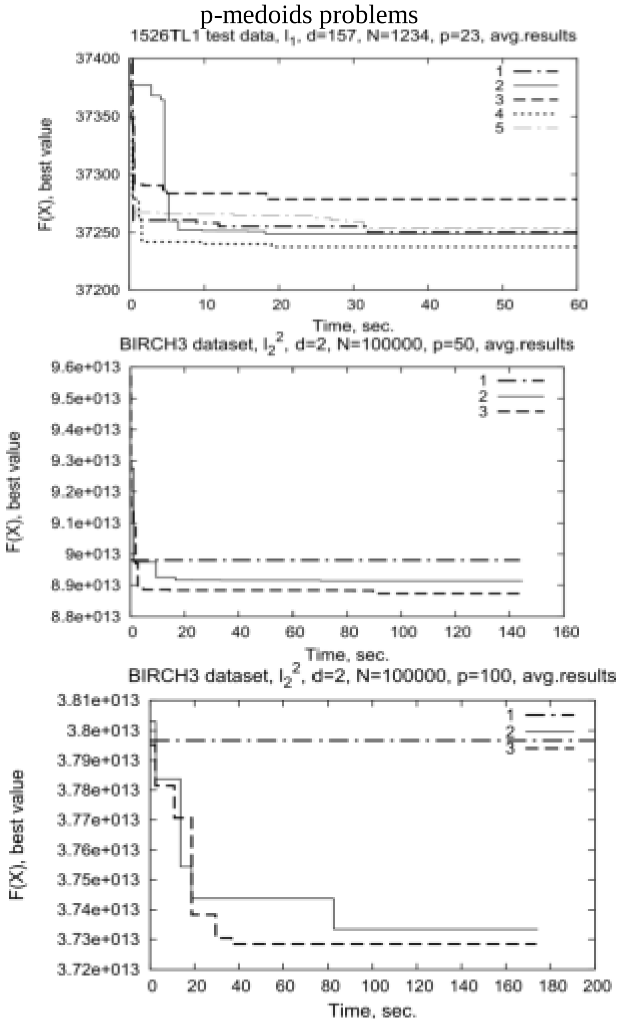
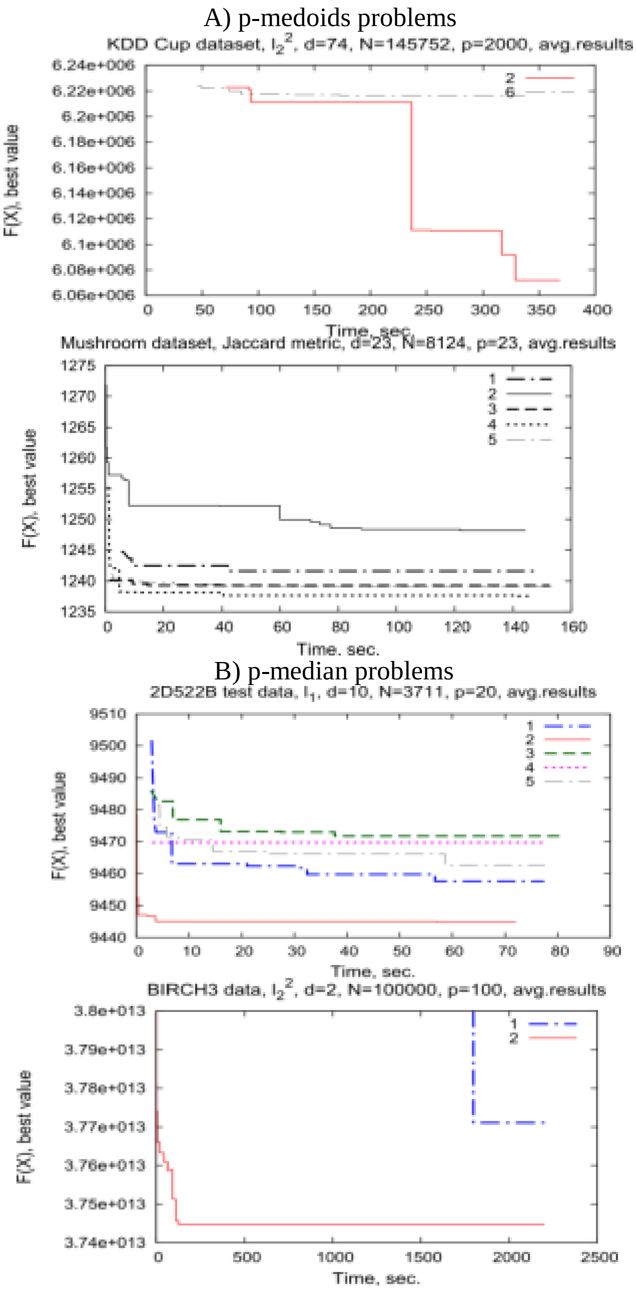


Fig. 1. Comparative results of algorithms. 1 – local SWAP search multistart, 2 – GA with greedy heuristic (PAM as a local search procedure), 3 – Algorithm 2 with σ=0 (in combination with SWAP search), 4 – Algorithm 2 with σ =0.2 (in combination with SWAP search), 5 – GA with fixed length subset recombination [43,29], 6 –PAM procedure multistart



A) p-medoids problems

B) p-median problems

Fig. 2. Comparative results of algorithms. 1 – local SWAP search multistart (j-means in the case of p-median problem), 2 – GA with greedy heuristic (PAM or ALA as a local search procedure), 3 – Algorithm 2 with σ=0 (in combination with SWAP search or j-means), 4 – Algorithm 2 with σ =0.2 (in combination with SWAP search or j-means), 5 – GA with fixed length subset recombination [43], 6 – ALA or PAM procedure multistart

Table 1. Comparative results for p-median problems, 30 runs.

Data set, its parameters	p and distance measure	Algorithm (see below)	Time, sec..	Average result	Std. deviation
Testing results for electronic chip 1526TL1, N=1234, d=120.	p=14, l_2^2	ALA multistart	15	150,124869801	0,384203928
		j-means multistart	15	150,533299444	0,598587789
		GA FL+ALA	15	149,954679652	0,172789313
		GA FL+j-means	15	151,280175427	0,982922979
		GA GH+ALA	15	149,78736565*	0,03157532*
		GA GH+j-means	15	151,082443691	0,654212395
	p=10, l_2^2	ALA multistart	15	198,375350991	0,018643710
		j-means multistart	15	198,426881563	0,044039446
		GA FL+ALA	15	198,377650812	0,024878118
		GA FL+j-means	15	198,450402498	0,032311263
		GA GH+ALA	15	198,359747028	$2\cdot10^{-14}$*
		GA GH+j-means	15	198,35421865*	0,0070903
	p=6, l_2^2	ALA multistart	15	362,70701636*	0*
		j-means multistart	15	362,70401636*	0*
		GA FL+ALA	15	362,70401636*	0*
		GA FL+j-means	15	362,704156850	0,000344112
		GA GH+ALA	15	362,704051312	0*
		GA GH+j-means	15	362,704051312	0*
UCI Mopsi Joensuu, N=6014, d=2.	p=10, l_2	ALA multistart	15	359,680203232	3,964320582
		j-means multistart	15	359,545287242	0,208756158
		GA FL+ALA	15	359,545250068	2,526439494
		GA FL+j-means	15	361,435624000	0,208770779
		GA GH+ALA	15	359,410460803	0,177992934
		GA GH+j-means	15	359,41036391*	0*
	p=4, l_2	ALA multistart	15	596,825210394	0,000000442
		j-means multistart	15	596,825217410	0,000004148
		GA FL+ALA	15	596,82520843*	0,000000388
		GA FL+j-means	15	596,825208927	0,000000574
		GA GH+ALA	15	596,825283111	0*
		GA GH+j-means	15	596,825283111	0*
BIRCH-3, N=100000, d=2.	p=100, l_2^2	ALA multistart	30	$3{,}7513245\cdot10^{13}$	116786778766
		j-means multistart	3000	$3{,}7711179\cdot10^{13}$	158613580914
		GA GH+ALA	30	$3{,}740432\cdot10^{13}$*	21699776156*
		GA GH+j-means	30	-	-
	p=50, l_2^2	GA FL+ALA	30	$9{,}0099578\cdot10^{13}$	9545892119
		GA FL+j-means	30	-	-
		GA GH+ALA	30	$8{,}902789\cdot10^{13}$*	0*
		GA GH+j-means	30	-	-
	p=20, l_2^2	GA FL+ALA	30	$3{,}303278\cdot10^{14}$*	0*
		GA FL+j-means	30	-	-
		GA GH+ALA	30	$3{,}3049972\cdot10^{14}$	0*
		GA GH+j-means	30	-	-

Notation: GA FL is the GA with fixed length subset recombination [43], GA GH is the GA with greedy heuristic. The best result is marked by “*”.

References

1. Ackermann, M.R. et al.: StreamKM: A Clustering Algorithm for Data Streams. J. Exp. Algorithmics **17**, Article 2.4 (2012), DOI:10.1145/2133803.2184450.

2. Aloise, D., Deshpande, A., Hansen, P. Popat, P.: NP-Hardness of Euclidean Sum-of-Squares Clustering. Machine Learning **75**, 245-249 (2009), DOI:10.1007/s10994-009-5103-0.
3. Alp, O., Erkut, E., Drezner, Z.: An Efficient Genetic Algorithm for the p-Median Problem. Annals of Operations Research **122**(1-4), 21–42 (2003), DOI 10.1023/A:1026130003508
4. Arthur, D., Vassilvitskii, D.: k-Means++:C The Advantages of Careful Seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete algorithms (SODA ’07), pp. 1027-1035. SIAM (2007).
5. Balcan, M.-F., Ehrlich, S., Liang, Y.: Distributed k-Means and k-Median Clustering on General Communication Topologies. Advances in Neural Information Processing Systems, pp. 1995-2003 (2013).
6. Bozkaya, B.A., Zhang, J., Erkut, E.: Genetic Algorithm for the p-Median Problem. In: Drezner, Z., Hamacher, H. [eds.].: Facility Location: Applications and Theory, pp.179-205, Springer, New York (2002).
7. Cooper, L.: An Extension of the Generalized Weber Problem. Journal of Regional Science **8**(2), 181-197 (1968).
8. Cooper, L.: Location-Allocation Problem. Operations Research **11**, 331-343 (1963).
9. Drezner, Z., Hamacher, H.: Facility Location: Applications and Theory, 460p., Springer-Verlag, Berlin (2004).
10. Drezner, Z., Wesolowsky, G.O.A.: Trajectory Method for the Optimization of the Multifacility Location Problem with lp Distances. Management Science **24**, 1507-1514 (1978).
11. Drineas, P., Frieze, A., Kannan, R., Vempala, S., Vinay, V.: Clustering Large Graphs via the Singular Value Decomposition. Machine learning **56**(1-3), 9-33 (1999).
12. Farahani, R., Hekmatfar, M. (eds.): Facility Location: Concepts, Models, Algorithms and Case Studies, 549 p., Springer-Verlag, Berlin-Heidelberg (2009).
13. Har-Peled, S., Mazudmar, S.: Coresets for k-Means and k-Median Clustering and their Applications. Proc. 36th Annu. ACM Sympos. Theory Comput., pp. 291-300 (2003).
14. Hakimi, S.L. Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. Operations Research **12**(3), 450-459 (1964).
15. Hansen P. Brimberg, J., Urosevic, D., Mladenovic, N.: Solving Large p-Median Clustering Problems by Primal-dual Variable Neighborhood Search. Data Mining and Knowledge Discovery **19**(3), 351-375 (2009).
16. Hansen, P., Mladenovic N.: J-Means: a New Local Search Heuristic for Minimum Sum of Squares Clustering. Pattern Recognition **34**(2), 405–413 (2001).
17. Hosage, C.M., Goodchild, M.F.: Discrete Space LocationAllocation Solutions from Genetic Algorithms. Annals of Operations Research **6**, 35-46 (1986).
18. Houck, C.R., Joines, J.A., Kay, G.M.: Comparison of Genetic Algorithms, Random Restart, and Two-Opt Switching for Solving Large Location-Allocation Problems. Computers and Operations Research **23**, 587-596 (1996).
19. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: an Introduction to Cluster Analysis. Wiley, 368 p., New York (1990).
20. Kazakovtsev, L.A., Antamoshkin, A.N., Masich, I.S.: Fast Deterministic Algorithm for EEE Components Classification. IOP Conf. Series: Materials Science and Engineering **94**, Article ID 012015, 10 p. (2015), DOI: 10.1088/1757-899X/04/1012015.
21. Kazakovtsev, L.A. Antamoshkin, N.A.: Genetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems. Informatica **38**(3), (2014).
22. Kazakovtsev, L.A., Antamoshkin, A.N. Greedy Heuristic Method for Location Problems. SibSAU Vestnik **16**(2), 317-325 (2015).

23. Kazakovtsev, L.A., Antamoshkin, A.N., Gudyma, M.N. Parallelnyi algoritm dlya p-mediannoy zadachi [Parallel Algorithm for p-Median Problem]. Sistemy Upravleniya I Informatsionnye Tekhnologii 52 (2.1), 124–128 (2013).
24. Kazakovtsev, L.A., Orlov, V.I., Stupina, A.A., Kazakovtsev, V.L.: Modied Genetic Algorithm with Greedy Heuristic for Continuous and Discrete p-Median Problems. Facta Universitatis, Series: Mathematics and Informatics **30** (1), 89-106 (2015).
25. Klastorin, T.D.: The p-Median Problem for Cluster Analysis: A Comparative Test Using the Mixture Model Approach. Management Science **31**(1), 84-95 (1985).
26. Kochetov, Yu. A.: Metody lokalnogo poiska dlya diskretnykh zadach razmescheniya [Local Search Methods for Discrete Location Problems]. Doctoral Thesis. 259 p., Sobolev Institute of Mathematics, Novosibirsk (2010).
27. Krishna, K., Murty, M.: Genetic K-Means Algorithm. IEEE Transaction on System, Man and Cybernetics - Part B **29**, 433-439 (1999).- Vol.29.
28. Liao, K., Guo, D.: A Clustering-Based Approach to the Capacitated Facility Location Problem. Transactions in GIS **12**(3), 323-339 (2008).
29. Lim, A., Xu, Z.: A Fixed-Length Subset Genetic Algorithm for the p-Median Problem. Lecture Notes in Computer Science **2724**, 1596-1597 (2003).
30. Lloyd, S.P.: Least Squares Quantization in PCM. IEEE Transactions on Information Theory **28**, 129-137 (1982).
31. Lucasius, C.B., Dane, A.D., Kateman, G.: On K-Medoid Clustering of Large Data Sets with the Aid of a Genetic Algorithm: Background, Feasibility and Comparison. Analytical Chimica Acta **282**, 647-669 (1993).
32. MacQueen, J.B.: Some Methods of Classification and Analysis of Multivariate Observations. Proceedings of the 5th Berkley Symposium on Mathematical Statistics and Probability **1**, 281–297 (1967).
33. Masuyama, S., Ibaraki, T., Hasegawa, T.: The Computational Complexity of the m-Center Problems on the Plane. The Transactions of the Institute of Electronics and Communication Engineers of Japan **64E**, 57-64 (1981).
34. Meira, L.A.A., Miyazawa, F.K.: A Continuous Facility Location Problem and Its Application to a Clustering Problem. Proceedings of the ACM symposium on Applied computing (SAC '08), pp.1826-1831, ACM, New York (2008), DOI:10.1145/1363686.1364126.
35. Mishra, N., Oblinger, D., Pitt, L.: Sublinear Time Approximate Clustering. 12th SODA, pp. 439–447 (2001).
36. Mladenovic, N., Brimberg, J., Hansen, P.: The p-Median Problem: A Survey of Metaheuristic Approaches. European Journal of Operational Research **179**(3), 927–939 (2007).
37. Moreno-Perez, J.A., Roda Garcia, J.L., Moreno-Vega, J.M.: A Parallel Genetic Algorithm for the Discrete p-Median Problem. Studies in Location Analysis 7, 131-141 (1994).
38. Muhlenbein, H., Shomisch, M., Born, J.: The parallel Genetic Algorithm as Function Optimizer. Proceedings of the Fourth Conference of Genetic Algorithms, pp.271-278, San Mateo,Morgan Kaufmann (1991).
39. Neema, M.N., Maniruzzaman, K.M., Ohgai, A.: New Genetic Algorithms Based Approaches to Continuous p-Median Problem. Netw. Spat. Econ. **11**, 83-99 (2011), DOI:10.1007/s11067-008-9084-5.
40. Resende, M.G.C.: Metaheuristic hybridization with Greedy Randomized Adaptive Search Procedures. In: Chen, Zh.-L., Raghavan, S. (eds.): TutORials in Operations Research, pp.295-319. INFORMS (2008).

41. Resende, M.G.C., Ribeiro, C.C., Glover, F., Marti, R.: Scatter Search and Path Relinking: Fundamentals, Advances, and Applications. In: Gendreau, M., Potvin, J.-Y. (eds.): Handbook of Metaheuristics, 2nd Edition, pp.87-107, Springer (2010).
42. Resende, M., Werneck, R.: On the Implementation of a Swap-Based Local Search Procedure for the p-Median Problem. Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments (ALENEX'03), pp.119-127, SIAM, Philadelphia (2003).
43. Sheng, W., Liu, X.: A Genetic k-Medoids Clustering Algorithm. Journal of Heuristics **12**, 447-466 (2006).
44. Sun, Zh., Fox, G., Gu, W., Li, Zh.: A Parallel Clustering Method Combined Information Bottleneck Theory and Centroid Based Clustering. The Journal of Supercomputing **69**(1), 452-467 (2014), DOI: 10.1007/s11227-014-1174-1.
45. Teitz, M.B., Bart, P. Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph. Operations Research 16, 955-961 (1968).
46. Zhang T., Ramakrishnan, R., Livny, M.: BIRCH: An Effcient Data Clustering Method for Very Large Databases. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD '96), pp. 103-114, ACM, New York (1996).