

HYBRID ALGORITHMS WITH ALTERNATIVE EMBEDDED LOCAL SEARCH SCHEMES FOR THE P-MEDIAN PROBLEM

*Natalya Rezova (1), Lev Kazakovtsev (1,2) *, Ivan Rozhnov (1,2), Predrag S. Stanimirovic (2,3), Guzel Shkaberina (1,2)*

⁽¹⁾ Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, Russia; ⁽²⁾ Siberian Federal University, Krasnoyarsk, Russia; ⁽³⁾ University of Nis, Faculty of Sciences and Mathematics, Nis, Serbia
Russia, Serbia

* Corresponding Author, e-mail: kazakovtsev.lev@bk.ru

Abstract: Progress in the development of location theory and clustering methods is mainly aimed at improving the performance of algorithms. We study the continuous p-median problem with the Euclidean metric. Various algorithmic combinations of the ALA procedure embedded into more sophisticated algorithmic combinations are used. Two repeated steps of the ALA algorithm solve the simplest discrete optimization problem, and then a series of central point search problems (Weber problem) for each cluster. A parallel implementation of the ALA algorithm is used. The paper studies the issue of choosing from two strategies: solve the Weber problem until accuracy ε is achieved at each iteration, or perform only one iteration of the Weiszfeld algorithm at each ALA iteration.

Key words: P-median problem, local search algorithms, Weber problem.

1. INTRODUCTION

The p-median facility location problem was set as a classic problem of the facility location theory. This problem is often considered as an unsupervised learning (clustering) model. For this problem, there are no known ways to obtain an exact solution in polynomial time if the data volume of the problem is of practical interest, and the efforts of researchers are focused at fast algorithms that lead to sub-optimal solutions having both comparatively high accuracy and speed.

The problem in a d-dimensional space is as follows. Given N points $A_1, \dots, A_N \in R^d$ and integer p , find a set $S = \{X_1, \dots, X_p\}$, $X_1, \dots, X_p \in R^d$, called a set of medians (centers) providing the minimum sum of distances [1]:

$$F(X_1, \dots, X_p) = F(S) = \sum_{i=1}^N w_i \min_{X \in \{X_1, \dots, X_p\}} L(A_i, X) \rightarrow \min_{X_1, \dots, X_p \in \mathbb{R}^d}, \quad (1)$$

where $w_i > 0$ ($i = \overline{1, N}$) are the weights, $L(\cdot, \cdot)$ denotes the distance. For an unweighted problem, $w_i = 1$ ($i = \overline{1, N}$).

An important special case of our problem is the Weber problem with a single sought median ($p = 1$) [2]:

$$f_{Web}(X) = \sum_{i=1}^N w_i L(X, A_i) \rightarrow \min_{X \in \mathbb{R}^d}. \quad (2)$$

Here, $X \in \mathbb{R}^d$, is a center (median) minimizing the weighted distance sum. For this problem, a simple iterative Weiszfeld procedure and its modifications guarantee finding a solution with the pre-set accuracy. In general case ($p > 1$), the Weiszfeld procedure can be used for finding a center in a subset (cluster) of points.

Generalized formulations of problem (1) may use various distance measures other than Euclidean. Moreover, $L(\cdot, \cdot)$ is non necessarily a metric. So, if we use the squared Euclidean distance $L(X, Y) = \sum_{l=1}^d (x_l - y_l)^2$ between two points $X = (x_1, \dots, x_d)$ and $Y = (y_1, \dots, y_d)$, then we get the k-means clustering problem [3]. For this case, there exists a very fast non-iterative way of solving the 1-median problem [4]:

$$x_j = \sum_{i=1}^N a_{i,j} / N \quad \forall j = \overline{1, d}, \quad (3)$$

where $X = (x_1, \dots, x_d)$ is the solution (called a centroid), $A_i = (a_{i,1}, \dots, a_{i,d}) \quad \forall i = \overline{1, N}$.

When using the square Euclidean distances, the center search requires much less computational resources. This fact increases the popularity of the k-means model. However, unlike the p-median, the k-means model is highly sensitive to the outliers (separately located points in a dataset) [5].

Being more computationally expensive, p-median problems are usually considered as cluster analysis models if the amount of data is relatively small or it is not necessary to obtain a solution with high accuracy. Parallelization of calculations on massive-parallel systems [6] enables us to use of p-median models with significant amounts of data.

Parallel implementation of simple local search procedures (e.g. Weiszfeld procedure) allows them to be used in the most complex algorithmic combinations, which provide the best accuracy, even for large datasets.

To solve the p-median problems, we use the algorithmic combinations of the most popular ALA procedure (ALA - Alternate Location-Allocation, Lloyd's algorithm) [7]. The ALA consists of two alternating steps of local search, each of

which is aimed at reducing the value of the objective function. Given the coordinates of the centers X_i , considering these centers as fixed, the first step of ALA, reassigns the points $A_1, \dots, A_N$ to the centers (medians). The data points are simply assigned to the closest median, and this guarantees the sum of the distances to be minimal with the fixed medians. In this process, clusters of data points are formed around the fixed medians. Given fixed clusters, the second step solves a Weber problem for each cluster. With Euclidean distance, the Weber problem is solved by the iterative Weiszfeld algorithm which enables us to obtain a result with the predefined accuracy ε and returns new coordinates X_i of the cluster center. Each iteration of this procedure is also aimed at the gradual improvement of the solution of the continuous optimization problem.

This paper studies the issue of choosing from two strategies: solve the Weber problem until a given accuracy ε is achieved at each iteration of the ALA algorithm (first approach), or perform only one iteration of the Weiszfeld algorithm at each iteration of the ALA (second approach). Organization of the paper is as follows. Section 2 provides an overview of the approaches known to solving p-median problems, including genetic algorithms (GA). Section 3 describes algorithmic combinations and a parallel implementation of the ALA algorithm for solving the p-median problem. Section 4 compares the two approaches by an experiment on large and medium sized datasets. Section 5 provides a conclusion.

2. KNOWN APPROACHES

For the first time, p-median problems were formulated for networks [8]. It was necessary to find p nodes (centers, medians) of the network such that the sums of weighted distances from the nodes to the nearest of p centers were minimal. This NP-hard [9] as well as the continuous problem. As the first attempts to solve them, exact methods were used [10], capable of solving very small problems. For larger problems, only heuristic approaches are considered as appropriate. The simplest approaches consist of a local search, in which the set of network nodes adjacent to the nodes of the current set of medians form a search neighborhood [11]. Algorithms developed within the graph theory [12] are suitable only for the smallest problems. For disconnected graphs, the authors of [13] presented a decomposition algorithm. The calculation results showed this approach was appropriate for rather large datasets.

The use of Lagrange relaxations makes it possible to obtain an approximate solution on medium size datasets [14]. Gokalp [15] used an iterative greedy approach based on compound local search. Many authors have considered various approaches to solving p-median problems [4, 16]. For continuous problems, Brimberg, Drezner, Mladenovic [17, 18] presented various local search algorithms, including Variable Neighborhood Search (VNS) algorithms. Reducing the amount of data [19] by randomly (or deterministically) selecting only a part of the original data set and using a result obtained with these sample data as an initial solution for

the complete data [14] is also rather efficient. Such approaches to aggregation, reducing the number of points [20] allow solving large-scale problems in a reasonable time, but lowers the accuracy of the solution. Local search algorithms and algorithmic combinations of the randomized and local search algorithms are investigated in many works. VNS algorithms [17] or agglomerative algorithms, as well as their combinations [21], often perform well. There are also widely presented initialization, including random filling and point distribution estimation [22]. However, more advanced algorithms make it possible to obtain the values of the objective function (the sum of distances) much better than local search methods and their combinations [21].

In [22], the authors introduce new AGGL neighbourhoods in addition to the traditionally used SWAP neighbourhoods, and compare the search efficiency. An AGGL neighborhood is defined as the set of possible outcomes after applying a greedy agglomerative operator to the current solution. The introduced AGGL neighbourhood have several parameters, and the authors investigate the dependence of algorithm efficiency on the parameter values. With properly tuned parameters, the search algorithms with AGGL neighborhoods return better distance sum values than the algorithms with SWAP neighbourhoods.

Algorithm 1, the Alternate Location-Allocation (ALA) procedure [7, 23], is capable of solving both k-means and p-median problems, was introduced by Lloyd [3]. This procedure consistently improves an intermediate solution, which leads to finding locally optimal solutions.

Algorithm 1: the Alternate Location-Allocation Procedure (ALA)

Given: Initial set of medians $X_1, \dots, X_p$.

1. For each median X_i , $i = \overline{1, p}$, determine its cluster $C_i \subset \{A_1, \dots, A_N\}$ as a subset of points with nearest X_i .
2. For each cluster C_i , $i = \overline{1, p}$, recalculate its median X_i , that is, solve the Weber problem: $X_i \leftarrow \operatorname{argmin}_X \sum_{Y \in C_i} w_i L(X, Y)$.
3. Repeat from step 1 if changes are made in steps 1 and 2.

Formally, this procedure is not a true local search algorithm: the search is not necessarily continued in a neighbourhood of an existing solution. However, it provides better results than true local optimization algorithms such as gradient descent.

The use of evolutionary algorithms (e.g. genetic ones) is a common approach to improve the local search results [5, 21]. They usually recombine the solutions obtained using the ALA procedure. To estimate the final value (local minimum) of the sum of distances (1), the GA at each iteration uses the ALA procedure or a local search algorithm. The mutation operator provides population variability by randomly changing the resulting solutions, and the crossover operator combines new sets of medians from existing ones. The general algorithmic framework for location and clustering problems was presented in [7].

3. ALGORITHMIC COMBINATIONS AND IMPLEMENTATIONS OF THE ALA ALGORITHM

Unlike other parallel architectures, CUDA (Compute Unified Device Architecture) enables us to run a many parallel threads. In the context of solving the p-median problem, we can organize a separate thread to process each of the points in the set $A_1, \dots, A_N \in R^d$. The papers [24, 25] present parallel (CUDA) versions of algorithms for the discrete problems. For the p-median problem with Euclidean distance, at step 2 of Algorithm 1, at each iteration, the Weiszfeld procedure is performed to solve the Weber problem with a given accuracy ε_1 . For the 1-median problem, at each iteration, the new median X' of the cluster $S' = \{A'_1, \dots, A'_{N'}\} \subset \{A_1, \dots, A_N\}$ of N' points is obtained from the current median X : $X' \leftarrow \sum_{i=1}^{N'} \frac{A'_i}{L(X, A'_i)} / \sum_{i=1}^{N'} \frac{1}{L(X, A'_i)}$. In the case $L(X, X') < \varepsilon_1$, the calculation stops. In most cases, this procedure converges comparatively fast, and only 10-50 iterations are needed to reach the hardware-defined precision of the double data type. The algorithm slows down if the median matches one of the data points. These situations are easy to work around:

$$X' \leftarrow \frac{\sum_{i \in \{1, N'\}, L(X, A'_i) < \varepsilon_2} \left[\frac{A'_i}{L(X, A'_i)} \right]}{\sum_{i \in \{1, N'\}, L(X, A'_i) < \varepsilon_2} \left[\frac{1}{L(X, A'_i)} \right]}, \quad (4)$$

where ε_2 ($\varepsilon_2 < \varepsilon_1$) is the minimum distance of distinguishability of two points.

The use of the Weiszfeld iterative algorithm in the ALA iterative algorithm (Algorithm 2), which itself is a part of the greedy agglomerative algorithm, makes it difficult to apply such an algorithmic construction efficiently to process large amounts of data.

Algorithm 2: The ALA algorithm for p-median problems

Given: Initial set of medians $X_1, \dots, X_p$.

1. For each median X_i , $i = \overline{1, p}$, determine its cluster $C_i \subset \{A_1, \dots, A_N\}$ as a subset of points with nearest X_i .
2. changed $\leftarrow 0$.
3. For each cluster C_i , $i = \overline{1, p}$, recalculate its median X_i , with the use of the Weiszfeld procedure:
 - 3.1 Calculate $X'_i \leftarrow \frac{\sum_{j \in C_i, L(X_i, A_j) > \varepsilon_2} \left[\frac{A_j}{L(X_i, A_j)} \right]}{\sum_{j \in C_i, L(X_i, A_j) > \varepsilon_2} \left[\frac{1}{L(X_i, A_j)} \right]}$;
 - 3.2 If $L(X'_i, X_i) > \varepsilon_1$ then changed $\leftarrow 1$;
 - 3.3 $X_i \leftarrow X'_i$;
 - 3.4 If changed then repeat from Step 3.1.
4. If changed = 1 go to Step 1.

5. Repeat from step 1 if changes are made in steps 1 and 2.

In our work, we explore the autonomous procedure ALA in the multi-start mode, its combinations with the SWAP neighbourhood search, as well as more complex four-level algorithmic combinations:

- level 1 - an algorithm that organizes a global search (evolutionary algorithm or multi-start mode) or extended local search (VNS);
- level 2 - greedy agglomerative procedure (Algorithm 3) (as a crossover and/or mutation operator in evolutionary algorithms or a neighborhood formation tool in VNS algorithms) [24];
- level 3 - ALA procedure;
- level 4 - Weiszfeld local search algorithm.

The greedy agglomerative heuristic procedure can be presented as a two-stage method. Firstly, two existing ("parent") solutions are united into a new intermediate solution with an excessive number of medians. In the second stage, medians at each iteration are eliminated so that the removal of the median leads to the least significant increase in the sum of distances (1), see Algorithm 3 [25].

Algorithm 3: Greedy Agglomerative Heuristic Procedure

Given: Initial number of medians K , final number of medians $k < K$, initial set of medians $S = \{X_1, \dots, X_K\}$.

1. Improve S with the Algorithm 1 (if possible).
2. while $K > k$
 - 2.1. For each $i' \in \{1, \overline{K}\}$
 - 2.1.1. Assign $S' \leftarrow S \setminus \{X_{i'}\}$.
 - 2.1.2. Calculate $F'_{i'} \leftarrow F(S')$ //where $F(\cdot)$ is the sum of distances (1)
 - 2.2: Select a subset S_{elim} of n_{elim} centers, $S_{elim} \subset S$, $|S_{elim}| = n_{elim}$ with the best (minimal) values of the sum of distances $F'_{i'}$. Here, $n_{elim} = \max\{1, 0.2 \cdot (|S| - k)\}$
 - 2.3. Obtain the new solution $S \leftarrow S \setminus S_{elim}$, $K \leftarrow |S|$, and apply Algorithm 1 to the new solution to improve it.
3. return Solution S .

Multilevel algorithms demonstrated their efficiency in achieving the most precise results for large problems. The last two levels, which account for the main computational load, are combined into one algorithm (Algorithm 2) for the two approaches indicated above:

1. the Weber problem is solved with a pre-set accuracy ε_1 at each iteration of Algorithm 2;
2. the Weiszfeld procedure is limited to one iteration (step 3.4 is excluded from Algorithm 2).

The Euclidean distance is calculated at each step of Algorithms 1, 2. For calculations, we used the CUDA implementation of Algorithm 2 described in [7]. This enabled us to use both the pure ALA algorithm and its combinations on large amounts of data.

4. RESULTS OF COMPUTATIONAL EXPERIMENTS

We used classical datasets from Clustering Basic Benchmark [26] and UCI Machine Learning Repository [27]: (a) Individual Household Electric Power Consumption (IHEPC, more than 2 million data vectors, 7 dimensions); (b) BIRCH3 - a synthetic dataset (100000 data vectors, 2-dimensional); (c) Mopsi-Joensuu - users location data (6014 data vectors, 2-dimensional).

The computing system included Intel Core 2 Duo E8400 CPU, 16GB RAM, NVIDIA GeForce GTX1050ti GPU with 4096MB RAM. For each algorithm, the experiment was carried out taking into account two approaches. Denote by W the maximum number of Weiszfeld iterations per iteration of the ALA algorithm. Then we will consider for the first approach $W = \infty$ and for the second approach $W = 1$, the best values of the sum of distances (for $W = 1$) are highlighted in bold.

We examined the following algorithms: GA with greedy agglomerative crossover (GA-FULL: the GA with greedy agglomerative heuristic procedures with full merge of the intermediate solutions; GA-ONE: genetic algorithm with greedy heuristics with partial union; GA-MIX: random selection of GA-ONE algorithms or GA-FULL [28]); ALA procedure in multistart mode; an algorithmic combination of the ALA and search in the SWAP neighborhood (centers are replaced by data points); GH-SWAP r : local search in SWAP r neighborhoods, $r = \overline{1, p}$ (only the best result for $r = 1, p$ is given); GH-VNS1, GH-VNS2, GH-VNS3: Variable Neighborhood Search algorithms with neighborhoods formed by application of AGGL r () procedure [11, 29]; GH_FIXGA: search algorithm in the neighborhood formed by the greedy agglomerative procedure AGGL r () with different values of the parameter r [29]; AGGL_EA: evolutionary algorithm for the continuous p-median problem with agglomerative mutation [29]. For algorithms launched in a multi-start mode (searching in SWAP and ALA), we fixed the best results in each of the attempts.

Iterations of the Weiszfeld procedure were performed with $\varepsilon_1 = 0,0001$, $\varepsilon_2 = 0,00001$.

The best (minimum), worst (maximum), average, and median results, as well as standard deviation of the achieved result were summarized after 30 runs of each algorithmic combination (Tables 1-3). In our experiments, we fixed the time limitation. For $W = 1$ the best values and of the result are highlighted in bold.

Table 1. Comparative results for Mopsi-Joensuu dataset ($N = 6014$ data vectors in R^2 , $p = 300$ centers, time limitation 5 second, GPU implementation)

Algorithm	Objective function value summarized after 30 attempts									
	Min (record)		Max (worst)		Avg.		Median value		Standard deviation	
	W=1	W=∞	W=1	W=∞	W=1	W=∞	W=1	W=∞	W=1	W=∞
ALA (multistart)	49,9726	51,6430	55,3898	60,2638	52,4256	56,6071	52,2383	56,8129	1,9112	2,7821
SWAP (multistart)	62,0784	66,8574	90,5706	101,0594	75,5619	82,2729	70,0661	82,1505	10,7874	11,8098
GH-VNS1	28,5495	28,0843	39,2323	36,0062	33,3201	31,6284	32,3284	31,1597	4,1079	2,3952
GH-VNS2	16,1216	15,8376	18,5159	17,4605	17,3117	16,7173	17,3121	16,6177	0,6781	0,5106
GH-VNS3	15,5378	15,6996	16,6759	17,9714	16,0280	16,4673	15,9405	16,2579	0,3638	0,6388
GA-MIX	14,9445	16,3247	18,6904	18,3947	16,5968	17,0487	16,3459	16,8917	1,0574	0,6718
GA-ONE	18,4014	22,3249	26,2358	27,8292	22,7824	26,1559	23,5659	26,6804	2,6833	1,8455
AGGL EA	15,4327	15,7486	16,1173	16,8788	15,8081	16,1363	15,8155	16,0567	0,2271	0,3269
GH-FIXGA-1	48,3750	58,6696	69,8820	66,5562	59,6361	63,2447	61,0443	64,1778	6,5204	2,5120
GH-FIXGA-10	17,1884	17,4939	20,4976	20,1792	18,5534	18,8867	18,4154	18,9420	1,0069	0,8653
GH-FIXGA-30	15,7788	15,9584	16,9611	18,0005	16,1921	16,6370	16,1463	16,5101	0,3748	0,5677
GH-FIXGA-100	15,7807	15,8271	16,3462	16,4538	16,0486	16,1029	16,0467	16,0716	0,1867	0,2079
GH-FIXGA-300	15,7588	15,6901	16,7225	17,2853	16,2404	16,3320	16,2048	16,3344	0,2721	0,4431
GH-SWAP -1	29,2655	24,3488	38,3482	38,5554	34,1496	34,0785	33,7013	34,9373	2,7471	3,9881
GH-SWAP -2	27,2080	22,0604	34,0894	36,1714	31,0791	30,6443	31,7233	31,5381	2,3382	4,0276
GH-SWAP -3	23,0133	24,6804	33,6960	35,8885	29,0518	30,4887	29,9189	29,9337	4,0898	2,8735
GH-SWAP -5	26,2961	28,1684	31,4489	32,1025	28,9201	30,2641	28,7159	30,5312	1,4250	1,2443
GH-SWAP -7	23,7825	27,8565	32,9865	32,2224	29,1032	29,5268	28,8428	29,0534	2,5392	1,3516
GH-SWAP -10	26,7194	26,1457	32,4875	31,3006	28,8536	29,3504	28,4061	30,2788	1,6256	1,8182
GH-SWAP -12	26,7754	27,6017	31,5995	33,8395	29,2821	30,6411	29,3832	30,0991	1,6789	1,7358
GH-SWAP -15	27,1562	26,6035	36,3557	33,1214	30,9735	30,7699	31,0377	31,6718	2,4673	2,0537

Table 2. Comparative results for BIRCH3 dataset ($N = 100000$ data vectors in R^2 , $p = 300$ centers, time limitation 10 seconds, GPU implementation)

Algorithm	Objective function value summarized after 30 attempts									
	Min (record)		Max (worst)		Avg.		Median value		Standard deviation	
	W=1	W=∞	W=1	W=∞	W=1	W=∞	W=1	W=∞	W=1	W=∞
ALA (multistart)	1,5578E+9	1,9541E+9	1,6401E+9	2,1389E+9	1,6120E+9	2,0512E+9	1,6198E+9	2,0558E+9	2,3692E+7	6,4530E+7
GH-VNS3	9,2641E+8	9,0535E+8	9,4720E+8	9,0790E+8	9,3598E+8	9,0650E+8	9,3326E+8	9,0641E+8	7,9172E+6	7,3533E+5
GA-MIX	9,0366E+8	9,0537E+8	1,1533E+9	9,6096E+8	9,9988E+8	9,1283E+8	9,9361E+8	9,0753E+8	7,5379E+7	1,6161E+7
GA-ONE	1,0120E+9	9,1736E+8	1,2376E+9	1,0527E+9	1,1126E+9	9,7213E+8	1,1499E+9	9,6483E+8	8,6288E+7	4,5332E+7
AGGL EA	9,0434E+8	9,0542E+8	9,0655E+8	9,0667E+8	9,0531E+8	9,0606E+8	9,0516E+8	9,0597E+8	6,5031E+5	4,7426E+5
GH-FIXGA-1	1,2783E+9	1,3745E+9	1,7599E+9	1,6536E+9	1,5375E+9	1,5001E+9	1,5463E+9	1,4814E+9	1,0240E+8	8,5486E+7
GH-FIXGA-10	1,0228E+9	9,0965E+8	1,1643E+9	9,4084E+8	1,0949E+9	9,2403E+8	1,0953E+9	9,2402E+8	4,5822E+7	8,2922E+6
GH-FIXGA-30	9,3992E+8	9,0492E+8	1,0582E+9	9,1089E+8	9,8002E+8	9,0695E+8	9,7408E+8	9,0624E+8	3,5867E+7	1,9954E+6
GH-FIXGA-100	9,0538E+8	9,0536E+8	9,2283E+8	9,0676E+8	9,1107E+8	9,0604E+8	9,0945E+8	9,0608E+8	5,5890E+6	4,1514E+5
GH-FIXGA-300	9,0464E+8	9,0579E+8	9,0590E+8	9,0665E+8	9,0516E+8	9,0616E+8	9,0506E+8	9,0608E+8	4,2781E+5	2,7772E+5
GH-SWAP -1	1,2573E+9	1,1281E+9	1,4561E+9	1,3042E+9	1,3728E+9	1,2018E+9	1,3844E+9	1,2070E+9	5,4890E+7	4,8017E+7
GH-SWAP -2	1,2167E+9	1,0938E+9	1,3691E+9	1,2651E+9	1,3156E+9	1,1546E+9	1,3444E+9	1,1508E+9	4,9485E+7	4,4411E+7
GH-SWAP -3	1,2386E+9	1,1347E+9	1,3795E+9	1,2115E+9	1,3152E+9	1,1650E+9	1,3092E+9	1,1547E+9	4,2649E+7	2,5743E+7
GH-SWAP -5	1,2940E+9	1,1246E+9	1,4584E+9	1,1938E+9	1,3772E+9	1,1746E+9	1,3813E+9	1,1787E+9	5,2215E+7	1,9192E+7
GH-SWAP -10	1,3232E+9	1,2091E+9	1,5201E+9	1,2762E+9	1,4421E+9	1,2390E+9	1,4744E+9	1,2371E+9	6,0378E+7	2,3234E+7
GH-SWAP -12	1,3690E+9	1,2474E+9	1,5352E+9	1,3299E+9	1,4588E+9	1,2806E+9	1,4626E+9	1,2690E+9	5,2955E+7	2,7419E+7
GH-SWAP -15	1,4137E+9	1,2579E+9	1,6064E+9	1,4187E+9	1,5110E+9	1,3222E+9	1,5005E+9	1,3289E+9	5,4951E+7	4,2663E+7

Table 3. Comparative results for IHEPC dataset ($N = 2075259$ data vectors in R^7 , $p = 100$ centers, time limitation 5 minutes, GPU implementation)

Algorithm	Objective function value summarized after 30 attempts									
	Min (record)		Max (worst)		Avg.		Median value		Standard deviation	
	$W=1$	$W=\infty$	$W=1$	$W=\infty$	$W=1$	$W=\infty$	$W=1$	$W=\infty$	$W=1$	$W=\infty$
ALA (multistart)	3,2872E+5	3,5548E+5	3,4076E+5	3,6320E+5	3,3717E+5	3,5953E+5	3,3849E+5	3,5910E+5	4,1679E+3	2,4206E+3
SWAP (multistart)	3,2015E+5	3,2520E+5	3,2513E+5	3,4904E+5	3,2317E+5	3,4183E+5	3,2405E+5	3,4470E+5	1,9798E+3	8,1523E+3
GH-VNS1	3,2700E+5	3,1789E+5	3,4430E+5	3,4360E+5	3,3878E+5	3,2499E+5	3,4074E+5	3,2203E+5	6,0829E+3	8,7228E+3
GH-VNS2	3,1227E+5	3,1225E+5	3,3941E+5	3,1333E+5	3,2723E+5	3,1286E+5	3,3524E+5	3,1280E+5	1,2223E+4	3,8585E+2
GH-VNS3	3,1147E+5	3,1206E+5	3,3714E+5	3,2998E+5	3,2288E+5	3,1701E+5	3,2227E+5	3,1346E+5	1,1446E+4	6,3986E+3
GA-1POINT	3,2220E+5	3,2117E+5	3,2785E+5	3,2765E+5	3,2515E+5	3,2447E+5	3,2475E+5	3,2425E+5	1,9887E+3	2,5540E+3
GA-FULL	3,1092E+5	3,1040E+5	3,1414E+5	3,1254E+5	3,1233E+5	3,1151E+5	3,1236E+5	3,1139E+5	9,7510E+2	7,5634E+2
GA-MIX	3,1137E+5	3,1093E+5	3,4361E+5	3,1454E+5	3,2492E+5	3,1224E+5	3,1394E+5	3,1233E+5	1,5434E+4	1,2841E+3
GA-ONE	3,1306E+5	3,1421E+5	3,4538E+5	3,2312E+5	3,2347E+5	3,1863E+5	3,1527E+5	3,1707E+5	1,4997E+4	3,1835E+3
GA-UNIFORM	3,2179E+5	3,1858E+5	3,2673E+5	3,2490E+5	3,2476E+5	3,2144E+5	3,2570E+5	3,2114E+5	1,8974E+3	2,3528E+3
AGGL EA	3,1100E+5	3,1127E+5	3,1280E+5	3,1230E+5	3,1197E+5	3,1185E+5	3,1184E+5	3,1199E+5	6,5310E+2	3,8951E+2
GH-FIXGA-1	3,1800E+5	3,1708E+5	3,4411E+5	3,2443E+5	3,3118E+5	3,2187E+5	3,2771E+5	3,2245E+5	1,1123E+4	2,3153E+3
GH-FIXGA-3	3,1445E+5	3,1407E+5	3,4229E+5	3,2001E+5	3,2958E+5	3,1689E+5	3,3641E+5	3,1684E+5	1,2737E+4	2,3149E+3
GH-FIXGA-10	3,1422E+5	3,1430E+5	3,1911E+5	3,1726E+5	3,1666E+5	3,1567E+5	3,1677E+5	3,1547E+5	1,9168E+3	1,0583E+3
GH-FIXGA-30	3,1301E+5	3,1214E+5	3,2528E+5	3,1496E+5	3,1599E+5	3,1352E+5	3,1397E+5	3,1334E+5	4,3823E+3	1,0224E+3
GH-FIXGA-100	3,1117E+5	3,1087E+5	3,1240E+5	3,1263E+5	3,1174E+5	3,1177E+5	3,1172E+5	3,1185E+5	4,7610E+2	6,5566E+2
GH-SWAP-1	3,3971E+5	3,2058E+5	3,4547E+5	3,2833E+5	3,4337E+5	3,2358E+5	3,4403E+5	3,2263E+5	2,0674E+3	2,7088E+3
GH-SWAP-2	3,1918E+5	3,2297E+5	3,4329E+5	3,2527E+5	3,3836E+5	3,2444E+5	3,4080E+5	3,2468E+5	8,5464E+3	8,5134E+2
GH-SWAP-3	3,2204E+5	3,1908E+5	3,4708E+5	3,3193E+5	3,3663E+5	3,2478E+5	3,4288E+5	3,2256E+5	1,0184E+4	4,6665E+3
GH-SWAP-5	3,3257E+5	3,1749E+5	3,4291E+5	3,2945E+5	3,4015E+5	3,2346E+5	3,4108E+5	3,2392E+5	3,5077E+3	3,6837E+3
GH-SWAP-10	3,3446E+5	3,2015E+5	3,4355E+5	3,3243E+5	3,4097E+5	3,2650E+5	3,4223E+5	3,2729E+5	3,0426E+3	3,8978E+3
GH-SWAP-15	3,4003E+5	3,2254E+5	3,4274E+5	3,3251E+5	3,4145E+5	3,2554E+5	3,4215E+5	3,2483E+5	1,2061E+3	3,2132E+3

Our experiments on problems of large and medium data volume in most cases demonstrate that in simple algorithms such as ALA or search in SWAP neighborhoods, the strategy with $W = 1$ almost always turns out to be the best. This cannot be unequivocally said about more complex algorithmic combinations, including algorithms with embedded greedy agglomerative procedures, where the efficiency of both strategies is comparable, although in most cases $W = 1$ still has an advantage in terms of the average and median achieved value of the sum of distances, at the same time, the strategy $W = \infty$ often provides more stable results (lower deviation of the result).

5. CONCLUSIONS

We studied the choice of two strategies for the p-median problem: solve the continuous Weber problem until a given accuracy ε is reached at each iteration of the algorithm ($W = \infty$, the first strategy), or perform only one iteration of the Weiszfeld algorithm at each iteration of the ALA ($W = 1$, second strategy). Computational experiments on problems of large and medium scale in most cases

have shown that the use of the $W = 1$ strategy is justified (gives the best value of the sum of distances) in simple algorithms such as ALA or search in the vicinity of SWAP. As for more complex algorithms, including algorithms with greedy agglomerative procedures, the choice of strategy is ambiguous. The strategy with $W = 1$ allows to achieve the best mean and median value of the sum of distances with an increase in speed time of the algorithm, while the strategy $W = \infty$ often gives more stable results (lower deviation of the sum of distances), while requiring significant computing resources. Thus, despite the higher efficiency of the algorithmic combinations of the ALA procedure with the embedded single iteration of the Weiszfeld procedure, in more complex algorithmic combinations based on the ALA procedure, this efficiency is not so evident.

ACKNOWLEDGMENTS

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

REFERENCES

- [1] Daskin, M.S., Maass, K.L. The p-Median Problem. In: Laporte, G., Nickel, S., Saldanha da Gama, F. (eds) *Location Science*. Springer, Cham. 2015. DOI: 10.1007/978-3-319-13111-5_2.
- [2] Wesolowsky, G. The Weber Problem: History and Perspectives. *Locat. Sci* vol.1, 1993, pp.5-23.
- [3] Celebi, M.E., Kingravi, H.A., Vela, P.A. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, Vol. 40, Issue 1, 2013, pp. 200-210. DOI: 10.1016/j.eswa.2012.07.021.
- [4] Farahani, R.Z., Hekmatfar, M. *Facility Location Concepts, Models, Algorithms and Case Studies*. Springer, Berlin/Heidelberg, Germany, 2009. DOI: 10.1007/978-3-7908-2151-2.
- [5] Ushakov, A. V., Vasilyev, I. L., Gruzdeva, T. V. A Computational Comparison of the p-Median Clustering and k-Means. *International Journal of Artificial Intelligence* vol.13, no.1, 2015, pp.229-242.
- [6] Firasta, N., Buxton, M., Jinbo, P., Nasri, K. and Kuo., S. Intel AVX: New Frontiers in Performance Improvements and Energy Efficiency, URL: <http://software.intel.com/> (Visited on 15.08.2023).
- [7] Kazakovtsev, L., Rozhnov, I., Shkaberina, G. Increasing Population Variability in Parallel Genetic Algorithms with a Greedy Crossover for Large-Scale p-Median Problems. *International Journal of Artificial Intelligence*, vol.19, no.2, 2021, pp.152-194.
- [8] Hakimi, S.L. Optimum locations of switching centers and the absolute centers and medians of a graph. *Oper. Res.* vol.12, 1964, pp.450-459. DOI: 10.1287/opre.12.3.450.

- [9] Jin, X., Han, J. K-Means Clustering. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. 2011. DOI: 10.1007/978-0-387-30164-8_425.
- [10] Rosing, K.E. An optimal method for solving the (generalized) multi-weber problem. Eur. J. Oper. Res. vol.58, 1992, pp.414-426. DOI: 10.1016/0377-2217(92)90072-H.
- [11] Rozhnov, I.P., Orlov, V.I., Kazakovtsev, L.A. VNS-Based algorithms for the centroid-based clustering problem. FACTA Univ. Ser. Math. Inform. vol.34, no.5, 2019, pp.957-972. DOI: 10.22190/FUMI1905957R.
- [12] Fathali, J., Rad, N.J., Sherbaf, S.R. The p-median and p-center problems on bipartite graphs. Iran. J. Math. Sci. Inf. vol.9, 2014, pp.37-43.
- [13] Agra, A., Cerdeira, J. O., Requejo, C. A Decomposition Approach for the p-Median Problem on Disconnected Graphs. Computers & Operations Research, vol.86, 2017, pp.79-85. DOI: 10.1016/j.cor.2017.05.006.
- [14] Avella, P., Boccia, M., Salerno, S., Vasilyev, I. An aggregation heuristic for large-scale p-median problem. Comput. Oper. Res. vol. 39, 2012, pp.1625-1632. DOI: 10.1016/j.cor.2011.09.016.
- [15] Gokalp, O. An Iterated Greedy Algorithm for the Obnoxious p-Median Problem. Eng. Appl. Artif. Intel vol.92, 2020, pp.103674. DOI: 10.1016/j.engappai.2020.103674.
- [16] Reese, J. Solution methods for the p-median problem: An annotated bibliography. Networks, vol.48 no.3, 2006, pp.125-142. DOI: 10.1002/net.20128.
- [17] Drezner, Z., Brimberg, J., Mladenovic, N., Salhi, S. Solving the planar p-median problem by variable neighborhood and con-centric searches. J. Glob. Optim, vol.63, 2015, pp.501-514. DOI: 10.1007/s10898-014-0183-1.
- [18] Mladenovic, N., Alkandari, A., Pei, J., Todosijevic, R., Pardalos, P.M. Less is more approach: basic variable neighborhood search for the obnoxious p -median problem. Int. Trans. Oper. Res. vol.27, 2019, pp.480-493. DOI: 10.1111/itor.12646.
- [19] Mu, V., Tong, D. On Solving Large p-Median Problems. Environment and Planning B: Urban Analytics and City Science, vol.47, no.6, 2020, pp.981-996. DOI: 10.1177/2399808319892598.
- [20] Irawan, C. A., Salhi, S. Solving Large p-Median Problems by a Multistage Hybrid Approach Using Demand Points Aggregation and Variable Neighbourhood Search. Journal of Global Optimization, vol.63, 2015, pp.537-554. DOI: 10.1007/s10898-013-0080-z.
- [21] Arthur, D., Vassilvitskii, S. k-Means++: The Advantages of Careful Seeding. *SODA'07 International Conference*, SIAM, New Orleans LA, USA, 2007, pp.1027-1035.
- [22] Kazakovtsev, L.A., Rozhnov, I.P. Comparative study of local search in SWAP and agglomerative neighbourhoods for the continuous p-median problem. IOP Conf. Ser.: Mater. Sci. Eng. vol.1047, 2021, pp.012079. DOI: 10.1088/1757-899X/1047/1/012079.

- [23] Kazakovtsev, L. A., Antamoshkin, A. N. Genetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems. *Informatica* vol.38, no.3, 2014, pp.229-240.
- [25] Kazakovtsev, L., Rozhnov, I., Shkaberina, G., Orlov, V. K-Means genetic algorithms with greedy genetic operators. *Math. Probl. Eng.* vol.202, 2020, pp.8839763. DOI: 10.1155/2020/8839763.
- [26] Drezner, Z. The Fortified Weiszfeld Algorithm for Solving the Weber Problem. *IMA Journal of Management Mathematics*, vol.26, 2013, pp.1-9. DOI: 10.1093/imaman/dpt019.
- [27] Clustering Basic Benchmark. URL: <http://cs.joensuu.fi/sipu/datasets/>. (Visited on 15.08.2023).
- [28] Dua, D., Graff, C. UCI Machine Learning Repository 2019. URL: <http://archive.ics.uci.edu/ml>. (Visited on 15.08.2023).
- [29] Kwedlo, W. Iwanowicz, P. Using Genetic Algorithm for Selection of Initial Cluster Centers for the k-Means Method. *ICAISC 2010 International Conference on Artificial Intelligence and Soft Computing*, 2010, pp.165-172. DOI: 10.1007/978-3-642-13232-2_20.
- [30] Kazakovtsev, L., Rozhnov, I., Shkaberina, G. Self-Configuring (1 + 1)-Evolutionary Algorithm for the Continuous p-Median Problem with Agglomerative Mutation. *Algorithms* vol.14, no.5, 2021, pp.130. DOI: 10.3390/a14050130.

Information about the authors:

Natalya Rezova is a Post-Graduate student of the Information and Management Systems of the Reshetnev State University of Science and Technology, Krasnoyarsk, Russia.

Lev Kazakovtsev is Doctor of Sciences, Professor. He is currently working as the Head of the Department of Operations Research and Systems Analysis of the Reshetnev State University of Science and Technology, Krasnoyarsk, Russian Federation.

Ivan Rozhnov is PhD, Professor at the Department of Digital Technologies of Management at the Siberian Federal University, Krasnoyarsk, Russia.

Predrag S. Stanimirovic currently works at the Faculty of Sciences and Mathematics (PMF), University of Niš. Predrag does research in Applied Mathematics, and Computer Science Education. Their current project is 'Generalized inverses and their applications'.

Guzel Shkaberina is currently working at the Information and Management Systems of the Reshetnev State University of Science and Technology, Krasnoyarsk, Russian Federation. She is PhD and Assistant Professor of Informatics.

Manuscript received on 30 October 2023