Chapter 7
Incremental Clustering Algorithms

7.1 Introduction

As we mentioned in Chap. 4, the clustering problem (4.3) is a nonsmooth global optimization problem and may have many local minimizers. Applying the conventional global optimization techniques is not always a good choice since they are time-consuming for solving such problems, particularly in large data sets. The local methods are fast, however, depending on the choice of starting cluster centers they may end up at the closest local minimizer. Therefore, the success of these methods in solving clustering problems heavily depends on the choice of initial centers.

Since the second half of 1980s, several algorithms have been introduced to choose *favorable* starting cluster centers for local search clustering algorithms, especially for the k-means algorithm [4, 14, 16, 19, 64, 190, 197]. In some of these algorithms, starting points are generated randomly using certain procedures. The use of the incremental approach allows us to choose good starting points in a deterministic way from different parts of the search space. The paper [106] is among the first introducing the incremental algorithm.

The existing incremental algorithms in cluster analysis can be divided, without any loss of generality, into the following classes:

- algorithms where new data points are added at each iteration and cluster centers are refined accordingly. Such algorithms are called *single pass incremental clustering algorithms;* and
- algorithms where clusters are built incrementally adding one cluster center at a time. This type of algorithms are called *sequential clustering algorithms*.

In the single pass incremental algorithms, new data points are presented as a sequence of items and can be examined only in a few passes (usually just one). At each iteration of these algorithms clusters are updated according to newly arrived

A. M. Bagirov et al., *Partitional Clustering via Nonsmooth Optimization*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-37826-4_7

data. These algorithms require limited memory and also limited processing time per item (see [130] and references therein).

In the second type of incremental algorithms, the data set is considered as static and clusters are computed incrementally. Such algorithms compute clusters step by step starting with one cluster for the whole data set and gradually adding one cluster center at each iteration [19, 26, 29, 142, 197]. In this book, we consider this type of incremental clustering algorithms.

There are following three optimization problems to be solved at each iteration of incremental clustering algorithms [229]:

- problem of finding a center of one cluster;
- auxiliary clustering problem, defined in (4.29), to obtain starting points for cluster centers; and
- clustering problem, given in (4.3), to determine all cluster centers.

In this chapter, we discuss different approaches for solving each of these problems. In Sect. 7.2, we describe how a center of one cluster can be found. The general incremental clustering algorithm is given in Sect. 7.3. This algorithm involves solving of the auxiliary clustering problem (4.29).

Since both the cluster and the auxiliary cluster functions are nonconvex they may have a large number of local minimizers. Therefore, having favorable starting points will help us to obtain either global or nearly global solutions to clustering problems. We describe the algorithm for finding such starting points for cluster centers in Sect. 7.4. This algorithm generates a set of starting points for the cluster centers, where the points guarantee the decrease of the cluster function at each iteration of the incremental algorithm. Section 7.5 presents the multi-start incremental clustering algorithm. This algorithm is an improvement of the general incremental algorithm that applies the algorithm for finding a set of starting cluster centers.

Finally, the incremental k-medians algorithm and the discussion on the decrease of its computational complexity are given in Sect. 7.6. This algorithm is a modification of the k-medians algorithm, where the latter algorithm is used at each iteration of the multi-start incremental algorithm to solve the clustering problem (4.3).

7.2 Finding a Center of One Cluster

In Chap. 5, the problem of finding a center of a cluster is formulated as an optimization problem. Considering a cluster C, the problem of finding its center $\boldsymbol{x} \in \mathbb{R}^n$ can be reformulated as follows:

$$\begin{cases} \text{minimize} & \varphi(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in \mathbb{R}^n, \end{cases} \tag{7.1}$$

where

$$\varphi(\boldsymbol{x}) = \frac{1}{|C|}\sum_{\boldsymbol{c}\in C} d_p(\boldsymbol{x}, \boldsymbol{c}).$$

If the similarity measure d_2 is used, then the centroid of the cluster C is the solution to the problem (7.1) which can be easily computed. If the distance function d_1 is applied, then according to Proposition 5.2 the median of the set C is a solution to this problem. This means that there is no need to solve the problem (7.1) when the similarity functions d_1 and d_2 are applied in the clustering problem.

Next, we consider the problem (7.1) when the function d_∞ is used. Unlike the functions d_1 and d_2, there is no explicit formula for finding a solution to this problem with the function d_∞, and one needs to apply some optimization methods to solve it. In this case, we have

$$\varphi(\boldsymbol{x}) = \frac{1}{|C|}\sum_{\boldsymbol{c}\in C} d_\infty(\boldsymbol{x}, \boldsymbol{c}),$$

and the subdifferential of the function φ at $\boldsymbol{x} \in \mathbb{R}^n$ is

$$\partial\varphi(\boldsymbol{x}) = \frac{1}{|C|}\sum_{\boldsymbol{c}\in C} \partial d_\infty(\boldsymbol{x}, \boldsymbol{c}),$$

where the subdifferential $\partial d_\infty(\boldsymbol{x}, \boldsymbol{c})$ is given in (4.10) and (4.11). Recall that the necessary and sufficient condition for a point $\boldsymbol{x}$ to be a minimum is $\boldsymbol{0} \in \partial\varphi(\boldsymbol{x})$.

For a moderately large number of points in the set C, the subdifferential $\partial\varphi(\boldsymbol{x})$ may have a huge number of extreme points and therefore, the computation of the whole subdifferential is not an easy task. To solve the problem (7.1) in this case, we can apply versions of the bundle method which are finite convergent for minimizing convex piecewise linear functions [32].

Another option is to use smoothing techniques to approximate the function d_∞ by the smooth functions to replace the problem (7.1) by the sequence of smooth optimization problems. Then we can apply any smooth optimization method to solve these problems.

7.3 General Incremental Clustering Algorithm

As we mentioned, the incremental approach provides an efficient way to generate starting cluster centers. In this section, we describe a general scheme of the *incremental clustering algorithm* (INC-CLUST) using the nonconvex nonsmooth optimization model of the clustering problem. Recall the clustering problem (4.3)

$$\begin{cases} \text{minimize} & f_k(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \in \mathbb{R}^{nk}, \end{cases} \tag{7.2}$$

where the function f_k, given in (4.4), is

$$f_k(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m} \min_{j=1,\dots,k} d_p(\boldsymbol{x}_j, \boldsymbol{a}_i). \tag{7.3}$$

We also recall the auxiliary clustering problem (4.29)

$$\begin{cases} \text{minimize} & \bar{f}_k(\boldsymbol{y}) \\ \text{subject to} & \boldsymbol{y} \in \mathbb{R}^n, \end{cases} \tag{7.4}$$

where the function $\bar{f}_k$, defined in (4.28), is

$$\bar{f}_k(\boldsymbol{y}) = \frac{1}{m}\sum_{i=1}^{m} \min\left\{r^i_{k-1}, d_p(\boldsymbol{y}, \boldsymbol{a}_i)\right\}, \tag{7.5}$$

and r^i_{k-1}, given in (4.27), is the distance between the data point $\boldsymbol{a}_i,\ i = 1, \dots, m$ and its cluster center:

$$r^i_{k-1} = \min_{j=1,\dots,k-1} d_p(\boldsymbol{x}_j, \boldsymbol{a}_i). \tag{7.6}$$

The general scheme of the INC-CLUST for solving the k-partition problem (7.2) is given in Fig. 7.1 and Algorithm 7.1.

Algorithm 7.1 Incremental clustering algorithm (INC-CLUST)

Input: Data set A and the number of clusters k to be computed.
Output: The l-partition of the set A with $l = 1, \dots, k$.

```
1: (Initialization) Compute the center $\boldsymbol{x}_1 \in \mathbb{R}^n$ of the set $A$. Set $l = 1$.
2: (Stopping criterion) Set $l = l + 1$. If $l > k$, then stop—the $k$-partition problem has been solved.
3: (Computation of the next cluster center) Find a starting point $\bar{\boldsymbol{y}} \in \mathbb{R}^n$ for the $l$th cluster center by solving the auxiliary clustering problem (7.4).
4: (Refinement of all cluster centers) Select $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_{l-1}, \bar{\boldsymbol{y}})$ as a starting point to solve the clustering problem (7.2) and find a solution $(\tilde{\boldsymbol{y}}_1, \dots, \tilde{\boldsymbol{y}}_l)$.
5: (Solution to the lth partition problem) Set $\boldsymbol{x}_j = \tilde{\boldsymbol{y}}_j,\ j = 1, \dots, l$ as a solution to the $l$th partition problem and go to Step 2.
```

Remark 7.1 Algorithm 7.1 in addition to the k-partition problem solves also all intermediate l-partition problems, where $l = 1, \dots, k - 1$.

Steps 3 and 4 are the most important steps of Algorithm 7.1, where both the auxiliary clustering problem (7.4) and the clustering problem (7.2) are solved. Since these problems are nonconvex they may have a large number of local minimizers.

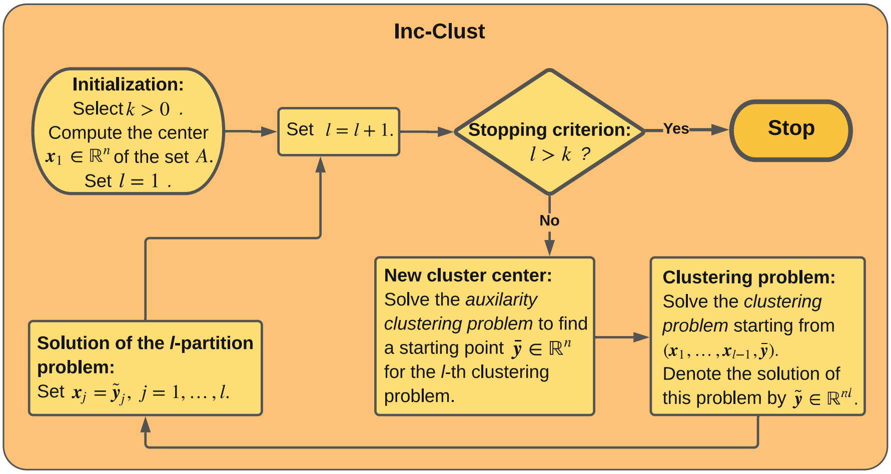

Inc-Clust
Initialization:
Select $k > 0$.
Compute the center
$x_1 \in \mathbb{R}^n$ of the set A.
Set $l = 1$.
Set $l = l + 1$.
Stopping criterion:
$l > k$?
Yes
Stop
No
New cluster center:
Solve the *auxilarity clustering problem* to find a starting point $\bar{y} \in \mathbb{R}^n$ for the l-th clustering problem.
Clustering problem:
Solve the *clustering problem* starting from $(x_1, \dots, x_{l-1}, \bar{y})$.
Denote the solution of this problem by $\tilde{y} \in \mathbb{R}^{nl}$.
Solution of the l-partition problem:
Set $x_j = \tilde{y}_j, \ j = 1, \dots, l$.

Fig. 7.1 Incremental clustering algorithm (Inc-Clust)

In the next section, we describe a special procedure to generate favorable starting points for solving these problems. Such an approach allows us to find high quality solutions to the clustering problem using local search methods.

7.4 Computation of Set of Starting Cluster Centers

In this section, first we describe an algorithm for finding starting points for solving the auxiliary clustering problem (7.4). We assume that for some $l > 1$, the solution $(x_1, \dots, x_{l-1})$ to the $(l-1)$-clustering problem is known. Consider the sets

$$\bar{S}_1 = \big\{ y \in \mathbb{R}^n : r^{a}_{l-1} \le d_p(y, a) \quad \text{for all} \quad a \in A \big\}, \quad \text{and} \tag{7.7}$$

$$\bar{S}_2 = \big\{ y \in \mathbb{R}^n : r^{a}_{l-1} > d_p(y, a) \quad \text{for some} \quad a \in A \big\}. \tag{7.8}$$

Here, $r^{a}_{l-1}, \ a \in A$ is defined by (4.27). It is obvious that cluster centers $x_1, \dots, x_{l-1} \in \bar{S}_1$. The set $\bar{S}_2$ contains all points $y \in \mathbb{R}^n$ which are not cluster centers and attract at least one point from the data set A.

Since the number $l-1$ of clusters is less than the number m of data points in the set A all points which are not cluster centers belong to the set $\bar{S}_2$ (because such points attract at least themselves) and therefore, the set $\bar{S}_2$ is not empty. Obviously

$$\bar{S}_1 \cap \bar{S}_2 = \emptyset \quad \text{and} \quad \bar{S}_1 \cup \bar{S}_2 = \mathbb{R}^n.$$

Figure 7.2 illustrates the sets $\bar{S}_1$ and $\bar{S}_2$ where the similarity measure d_2 is applied to find cluster centers. There are three clusters in this figure. Their centers are shown by “red” circles. The set $\bar{S}_2$ consists of all points inside three balls except cluster

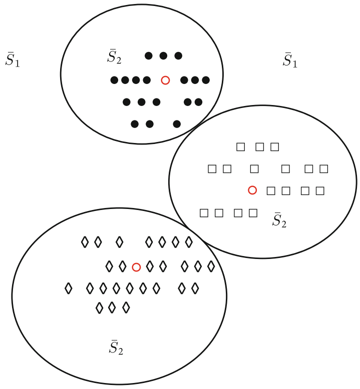


Fig. 7.2 Illustration of sets $\bar{S}_1$ and $\bar{S}_2$

centers and the set $\bar{S}_1$ contains three cluster centers and the part of the space outside balls.

Note that

$$\bar{f}_l(\mathbf{y}) \le \frac{1}{m}\sum_{\mathbf{a}\in A} r_{l-1}^{\mathbf{a}} \quad \text{for all} \quad \mathbf{y}\in\mathbb{R}^n, \quad \text{and}$$

$$\bar{f}_l(\mathbf{y}) = f_{l-1}(\mathbf{x}_1,\dots,\mathbf{x}_{l-1}) = \frac{1}{m}\sum_{\mathbf{a}\in A} r_{l-1}^{\mathbf{a}} \quad \text{for all} \quad \mathbf{y}\in\bar{S}_1.$$

This means that the lth auxiliary cluster function $\bar{f}_l$ is constant on the set $\bar{S}_1$, and any point from this set is a global maximizer of this function. In general, a local search method terminates at any of these points. Therefore, starting points for solving the auxiliary clustering problem (7.4) should not be chosen from the set $\bar{S}_1$.

We introduce a special procedure which allows one to select starting points from the set $\bar{S}_2$. Take any $\mathbf{y}\in\bar{S}_2$ and consider the sets $B_i(\mathbf{y}),\ i = 1, 2, 3$ defined in (4.30). Then the set A can be divided into two subsets $\bar{B}_{12}(\mathbf{y})$ and $\bar{B}_3(\mathbf{y})$, where

$$\bar{B}_{12}(\mathbf{y}) = B_1(\mathbf{y}) \cup B_2(\mathbf{y}) \quad \text{and} \quad \bar{B}_3(\mathbf{y}) = B_3(\mathbf{y}). \tag{7.9}$$

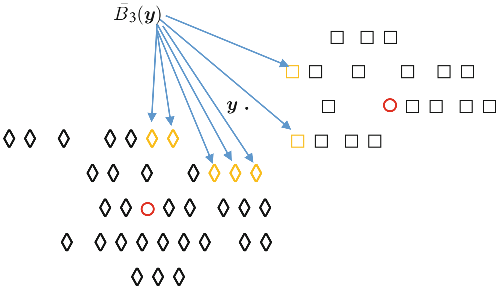
The set $\bar{B}_3(\mathbf{y})$ contains all data points $\mathbf{a}\in A$ which are closer to the point $\mathbf{y}$ than to their cluster centers, and the set $\bar{B}_{12}(\mathbf{y})$ contains all other data points. Since $\mathbf{y}\in\bar{S}_2$ the set $\bar{B}_3(\mathbf{y}) \ne \emptyset$. Furthermore,

$$\bar{B}_{12}(\mathbf{y}) \cap \bar{B}_3(\mathbf{y}) = \emptyset \quad \text{and} \quad A = \bar{B}_{12}(\mathbf{y}) \cup \bar{B}_3(\mathbf{y}).$$

Figure 7.3 depicts the set $\bar{B}_3(\mathbf{y})$ for a given $\mathbf{y}$ (black ball). There are two clusters in this data set and their centers are shown by “red” circles. The set $\bar{B}_3(\mathbf{y})$ contains all “yellow” data points and the set $\bar{B}_{12}(\mathbf{y})$ contains the rest of the data set.

At a point $\mathbf{y}\in\mathbb{R}^n$ using the sets $\bar{B}_{12}(\mathbf{y})$ and $\bar{B}_3(\mathbf{y})$, the lth auxiliary cluster function $\bar{f}_l$ can be written as

Fig. 7.3 Illustration of sets $\bar{B}_{12}(\boldsymbol{y})$ and $\bar{B}_3(\boldsymbol{y})$



$$\bar{f}_l(\boldsymbol{y}) = \frac{1}{m}\Big(\sum_{\boldsymbol{a}\in\bar{B}_{12}(\boldsymbol{y})} r_{l-1}^{\boldsymbol{a}} + \sum_{\boldsymbol{a}\in\bar{B}_3(\boldsymbol{y})} d_p(\boldsymbol{y},\boldsymbol{a})\Big).$$

The difference between the values of $\bar{f}_l(\boldsymbol{y})$ and $f_{l-1}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{l-1})$ is

$$z_l(\boldsymbol{y}) = \frac{1}{m}\sum_{\boldsymbol{a}\in\bar{B}_3(\boldsymbol{y})} \Big(r_{l-1}^{\boldsymbol{a}} - d_p(\boldsymbol{y},\boldsymbol{a})\Big),$$

which can be rewritten as

$$z_l(\boldsymbol{y}) = \frac{1}{m}\sum_{\boldsymbol{a}\in A} \max \Big\{0, r_{l-1}^{\boldsymbol{a}} - d_p(\boldsymbol{y},\boldsymbol{a})\Big\}. \tag{7.10}$$

The difference $z_l(\boldsymbol{y})$ shows the decrease of the value of the lth cluster function f_l comparing with the value $f_{l-1}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{l-1})$ if the points $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{l-1},\boldsymbol{y}$ are chosen as the cluster centers for the lth clustering problem.

It is reasonable to choose a point $\boldsymbol{y} \in \mathbb{R}^n$ that provides the largest decrease $z_l(\boldsymbol{y})$ of the clustering function as the starting point for minimizing the auxiliary clustering function. Since it is not easy to choose such a point from the whole space $\mathbb{R}^n$ we restrict ourselves to the data set A.

If a data point $\boldsymbol{a} \in A$ is a cluster center, then this point belongs to the set $\bar{S}_1$, otherwise it belongs to the set $\bar{S}_2$. Therefore, we choose points $\boldsymbol{y}$ from the set $\bar{A}_0 = A \setminus \bar{S}_1$. Obviously, $\bar{A}_0 \neq \emptyset$. Take any $\boldsymbol{y} = \boldsymbol{a} \in \bar{A}_0$, compute $z_l(\boldsymbol{a})$ and define the number

$$z_{\max}^1 = \max_{\boldsymbol{a}\in\bar{A}_0} z_l(\boldsymbol{a}). \tag{7.11}$$

The number $z_{\max}^1$ represents the largest decrease of the cluster function which can be provided by any data point. Let $\gamma_1 \in [0,1]$ be a given number. Compute the following subset of $\bar{A}_0$:

$$\bar{A}_1 = \big\{\boldsymbol{a} \in \bar{A}_0 : \ z_l(\boldsymbol{a}) \geq \gamma_1 z_{\max}^1\big\}. \tag{7.12}$$

The set $\bar{A}_1$ contains all data points that provide the decrease of the cluster function no less than the threshold $\gamma_1 z^1_{\max}$. This set is obtained from the set $\bar{A}_0$ by removing data points that do not provide sufficient decrease of the cluster function. Apparently, $\bar{A}_1 \neq \emptyset$ for any $\gamma_1 \in [0, 1]$. If $\gamma_1 = 0$, then $\bar{A}_1 = \bar{A}_0$ and if $\gamma_1 = 1$, then the set $\bar{A}_1$ contains data points providing the largest decrease $z^1_{\max}$.

For each point $\boldsymbol{a} \in \bar{A}_1$ compute the set $\bar{B}_3(\boldsymbol{a})$ and its center $\boldsymbol{c}(\boldsymbol{a})$. Replace the point $\boldsymbol{a}$ by $\boldsymbol{c}(\boldsymbol{a})$ since the center $\boldsymbol{c}(\boldsymbol{a})$ is a better representative of the set $\bar{B}_3(\boldsymbol{a})$ than the point $\boldsymbol{a}$. If the similarity measure d_p is defined using the L_2-norm, then $\boldsymbol{c}(\boldsymbol{a})$ is the centroid of the set $\bar{B}_3(\boldsymbol{a})$. In other cases, $\boldsymbol{c}(\boldsymbol{a})$ is found as a solution to the problem (7.1) where

$$\varphi(\boldsymbol{x}) = \frac{1}{|\bar{B}_3(\boldsymbol{a})|} \sum_{\boldsymbol{b} \in \bar{B}_3(\boldsymbol{a})} d_p(\boldsymbol{x}, \boldsymbol{b}).$$

Let

$$\bar{A}_2 = \left\{ \boldsymbol{c} \in \mathbb{R}^n : \text{ there exists } \boldsymbol{a} \in \bar{A}_1 \text{ such that } \boldsymbol{c} = \boldsymbol{c}(\boldsymbol{a}) \right\}$$

be the set of such solutions. It is obvious that $\bar{A}_2 \neq \emptyset$. For each $\boldsymbol{c} \in \bar{A}_2$, compute the number $z_l(\boldsymbol{c})$ using (7.10) and find the number

$$z^2_{\max} = \max_{\boldsymbol{c} \in \bar{A}_2} z_l(\boldsymbol{c}). \tag{7.13}$$

The number $z^2_{\max}$ represents the largest value of the decrease

$$f_{l-1}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}) - f_l(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}, \boldsymbol{c})$$

among all centers $\boldsymbol{c} \in \bar{A}_2$.

For a given number $\gamma_2 \in [0, 1]$, define the following subset of $\bar{A}_2$:

$$\bar{A}_3 = \left\{ \boldsymbol{c} \in \bar{A}_2 : \ z_l(\boldsymbol{c}) \geq \gamma_2 z^2_{\max} \right\}. \tag{7.14}$$

The set $\bar{A}_3$ contains all points $\boldsymbol{c} \in \bar{A}_2$ that provide the decrease of the cluster function no less than the threshold $\gamma_2 z^2_{\max}$. This set is obtained from the set $\bar{A}_2$ by removing centers which do not provide the sufficient decrease of the cluster function. It is clear that the set $\bar{A}_3 \neq \emptyset$ for any $\gamma_2 \in [0, 1]$. If $\gamma_2 = 0$, then $\bar{A}_3 = \bar{A}_2$ and if $\gamma_2 = 1$, then the set $\bar{A}_3$ contains only centers $\boldsymbol{c}$ providing the largest decrease of the cluster function f_l.

All points from the set $\bar{A}_3$ are considered as starting points for solving the auxiliary clustering problem (7.4). Since all data points are used for the computation of the set $\bar{A}_3$, it contains starting points from different parts of the data set. Such a strategy allows us to find either global or nearly global solutions to the problem (7.2) (as well as to the problem (7.4)) using local search methods.

Applying a local search algorithm, the auxiliary clustering problem (7.4) is solved using starting points from $\bar{A}_3$. A local search algorithm generates the same number of solutions as the number of starting points. The set of these solutions is denoted by $\bar{A}_4$. This set is a non-empty subset of the set of stationary points of the auxiliary cluster function $\bar{f}_l$.

A local search algorithm starting from different points may arrive to the same stationary point or stationary points which are close to each other. To identify such stationary points we define a tolerance $\varepsilon > 0$. If the distance between any two points from the set $\bar{A}_4$ is less than this tolerance, then we keep a point with the lower value of the function $\bar{f}_l$ and remove another point from the set $\bar{A}_4$.

Next, we define

$$\bar{f}_l^{\min} = \min_{\boldsymbol{y} \in \bar{A}_4} \bar{f}_l(\boldsymbol{y}). \tag{7.15}$$

The number $\bar{f}_l^{\min}$ is the lowest value of the auxiliary cluster function $\bar{f}_l$ over the set $\bar{A}_4$. Let $\gamma_3 \in [1, \infty)$ be a given number. Introduce the following set:

$$\bar{A}_5 = \left\{\boldsymbol{y} \in \bar{A}_4 : \bar{f}_l(\boldsymbol{y}) \le \gamma_3 \bar{f}_l^{\min}\right\}. \tag{7.16}$$

The set $\bar{A}_5$ contains all stationary points where the value of the function $\bar{f}_l$ is no more than the threshold $\gamma_3 \bar{f}_l^{\min}$. Note that the set $\bar{A}_5 \neq \emptyset$. If $\gamma_3 = 1$, then $\bar{A}_5$ contains the best local minimizers of the function $\bar{f}_l$ obtained using starting points from the set $\bar{A}_3$. If γ_3 is sufficiently large, then $\bar{A}_5 = \bar{A}_4$. Points from the set $\bar{A}_5$ are used as a set of starting points for the lth cluster center to solve the lth clustering problem (7.2).

Summarizing all described above, the algorithm for finding starting points to solve the problem (7.2) proceeds as follows [228].

Algorithm 7.2 allows us to use more than one starting point to solve the clustering problem (7.2) in Step 4 of Algorithm 7.1. Moreover, these points always guarantee the decrease of the cluster function value at each iteration of the incremental algorithm and are distinct from each other in the search space. Such an approach

Algorithm 7.2 Finding set of starting points for the lth cluster center

Input: Data set A and the solution $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1})$ to the $(l-1)$-clustering problem, $l \ge 2$.
Output: Set of starting points for the lth cluster center.

```
(Initialization) Select numbers $\gamma_1, \gamma_2 \in [0, 1]$ and $\gamma_3 \in [1, \infty)$.
Compute $z^1_{\max}$ using (7.11) and the set $\bar{A}_1$ using (7.12).
Compute $z^2_{\max}$ using (7.13) and the set $\bar{A}_3$ using (7.14).
Compute the set $\bar{A}_4$ of stationary points of the auxiliary clustering problem (7.4) applying a
  local search algorithm and using starting points from the set $\bar{A}_3$.
Compute $\bar{f}_l^{\min}$ using (7.15) and the set $\bar{A}_5$ using (7.16). $\bar{A}_5$ is the set of starting points for the
  $l$th cluster center.
```

Algorithm 7.3 Multi-start incremental clustering algorithm (MSInc-Clust)

```
Input: Data set $A$ and the number of clusters $k$ to be computed.
Output: The $l$-partition of the set $A$ with $l = 1, \dots, k$.

1: (Initialization) Compute the center $\mathbf{x}_1 \in \mathbb{R}^n$ of the set $A$. Set $l = 1$.
2: (Stopping criterion) Set $l = l + 1$. If $l > k$, then stop—the $k$-partition problem has been
   solved.
3: (Computation of a set of starting points for the lth cluster center) Apply Algorithm 7.2 to
   compute the set $\bar{A}_5$ defined by (7.16).
4: (Computation of a set of cluster centers) For each $\bar{\mathbf{y}} \in \bar{A}_5$, select $(\mathbf{x}_1, \dots, \mathbf{x}_{l-1}, \bar{\mathbf{y}})$ as a starting
   point to solve the $l$th clustering problem (7.2) and find its solution $(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l)$. Denote by $\bar{A}_6$
   a set of all such solutions.
5: (Computation of the best solution) Compute
      $$f_l^{\min} = \min \Big\{ f_l(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l) : (\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l) \in \bar{A}_6 \Big\},$$
   and the collection of cluster centers $(\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_l)$ such that
      $$f_l(\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_l) = f_l^{\min}.$$
6: (Solution to the lth partition problem) Set $\mathbf{x}_j = \tilde{\mathbf{y}}_j, \ j = 1, \dots, l$ as a solution to the $l$th
   partition problem and go to Step 2.
```

allows us to apply local search methods to obtain a high quality solution to the global optimization problem (7.2).

7.5 Multi-Start Incremental Clustering Algorithm

In this section, we present the *multi-start incremental clustering algorithm* (MSInc-Clust) for solving the problem (7.2). This algorithm is an improvement of Algorithm 7.1 where in Step 3, Algorithm 7.2 is applied. Similar to Algorithm 7.1, the MSInc-Clust builds clusters dynamically adding one cluster center at a time by solving the auxiliary clustering problem (7.4).

The MSInc-Clust applies Algorithm 7.2 to compute a set of starting cluster centers. Using these centers as initial points, the lth clustering problem (7.2) is solved ($l = 2, \dots, k$). Then a solution with the least cluster function value, defined in (7.3), is accepted as the solution to the clustering problem. The flowchart of the MSInc-Clust is given in Fig. 7.4 and its step by step description is presented in Algorithm 7.3.

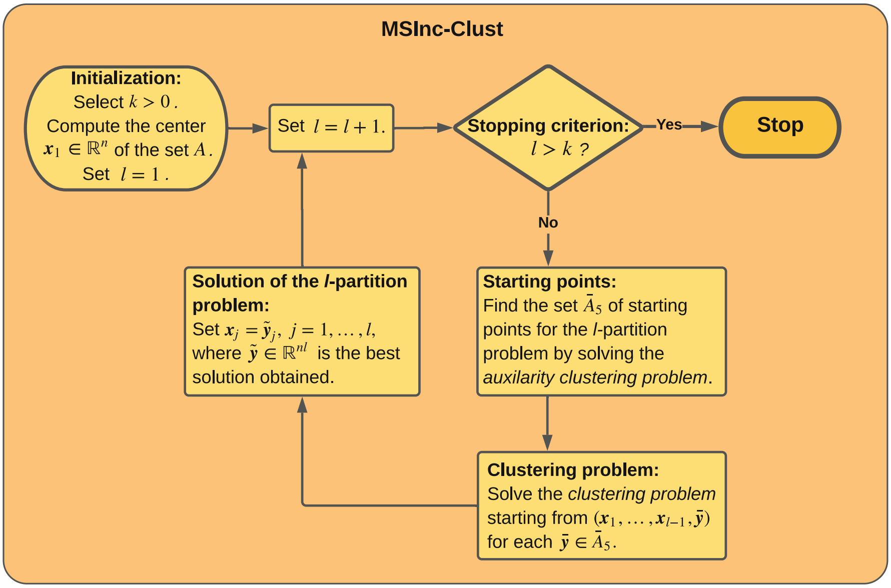

MSInc-Clust
Initialization:
Select $k > 0$.
Compute the center
$\boldsymbol{x}_1 \in \mathbb{R}^n$ of the set A.
Set $l = 1$.
Set $l = l + 1$.
Stopping criterion:
$l > k$?
Yes
Stop
No
Starting points:
Find the set $\bar{A}_5$ of starting
points for the l-partition
problem by solving the
auxilarity clustering problem.
Clustering problem:
Solve the *clustering problem*
starting from $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}, \bar{\boldsymbol{y}})$
for each $\bar{\boldsymbol{y}} \in \bar{A}_5$.
Solution of the l-partition
problem:
Set $\boldsymbol{x}_j = \tilde{\boldsymbol{y}}_j,\ j = 1, \ldots, l,$
where $\tilde{\boldsymbol{y}} \in \mathbb{R}^{nl}$ is the best
solution obtained.

Fig. 7.4 Multi-start incremental clustering algorithm (MSINC-CLUST)

Remark 7.2 Similar to Algorithm 7.1, this algorithm solves all intermediate l-partition problems ($l = 1, \ldots, k - 1$) in addition to the k-partition problem. However, Algorithm 7.1 can find only stationary points of the clustering problem, while Algorithm 7.3 is able to find either global or nearly global solutions.

Note that the most important steps in Algorithm 7.3 are Step 3, where the auxiliary clustering problem (4.29) is solved to find starting points, and Step 4, where the clustering problem (7.2) is solved for each starting point. To solve these problems, we will introduce different algorithms in this and the following two chapters.

7.6 Incremental k-Medians Algorithm

In this section, we design the *incremental k-medians algorithm* (IKMED) as an application of Algorithm 7.3. The k-medians algorithm (Algorithm 5.4), presented in Chap. 5, is simple and easy to implement. However, this algorithm is sensitive to the choice of starting points and finds only local solutions that can be significantly different from the global solution in large data sets. The IKMED overcomes these

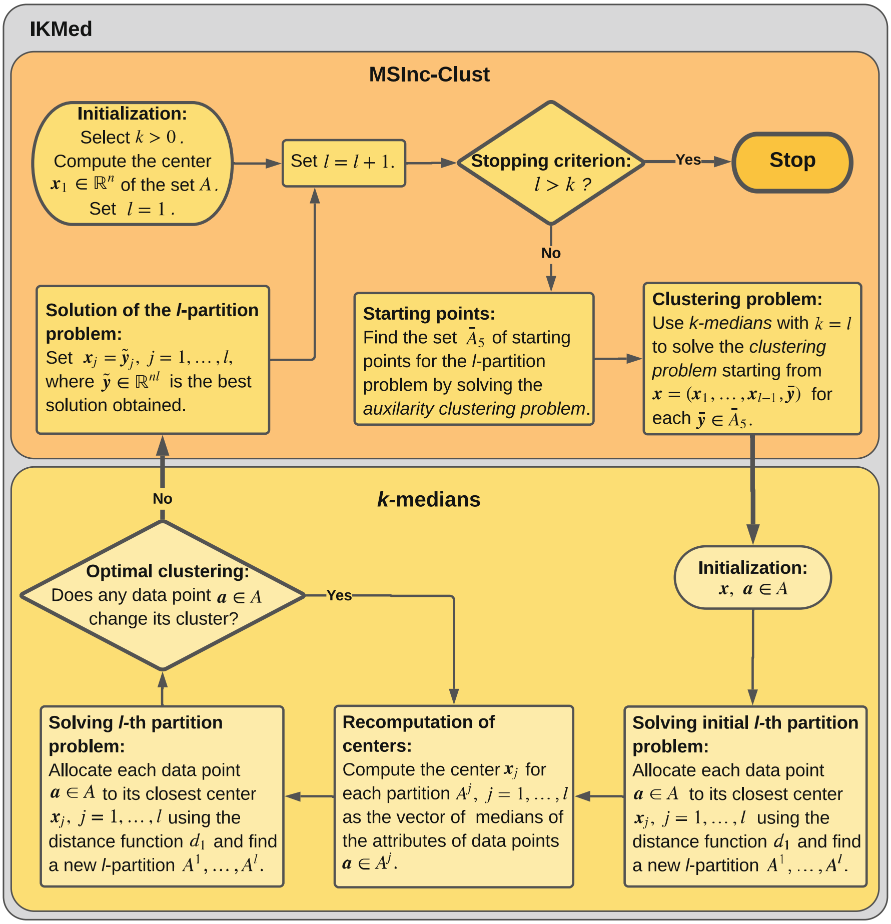

IKMed
MSInc-Clust
Initialization: Select $k > 0$. Compute the center $x_1 \in \mathbb{R}^n$ of the set A. Set $l = 1$.
Set $l = l + 1$.
Stopping criterion: $l > k$?
Yes
Stop
No
Solution of the l-partition problem: Set $x_j = \tilde{y}_j,\ j = 1, \dots, l$, where $\tilde{y} \in \mathbb{R}^{nl}$ is the best solution obtained.
Starting points: Find the set $\bar{A}_5$ of starting points for the l-partition problem by solving the auxilarity clustering problem.
Clustering problem: Use k-medians with $k = l$ to solve the clustering problem starting from $x = (x_1, \dots, x_{l-1}, \bar{y})$ for each $\bar{y} \in \bar{A}_5$.
No
k-medians
Optimal clustering: Does any data point $a \in A$ change its cluster?
Yes
Initialization: $x,\ a \in A$
Solving l-th partition problem: Allocate each data point $a \in A$ to its closest center $x_j,\ j = 1, \dots, l$ using the distance function d_1 and find a new l-partition $A^1, \dots, A^l$.
Recomputation of centers: Compute the center x_j for each partition $A^j,\ j = 1, \dots, l$ as the vector of medians of the attributes of data points $a \in A^j$.
Solving initial l-th partition problem: Allocate each data point $a \in A$ to its closest center $x_j,\ j = 1, \dots, l$ using the distance function d_1 and find a new l-partition $A^1, \dots, A^l$.

Fig. 7.5 Incremental k-medians algorithm (IKMED)

drawbacks by applying Algorithm 7.2. Characteristically for k-medians, the distance function d_1 is used to define the similarity measure in the IKMED. Fig. 7.5 illustrates the flowchart of this algorithm.

The IKMED first calculates the center of the whole data set as its median. Then it applies Algorithm 7.2 to compute the set of initial cluster centers by solving the auxiliary clustering problem (7.4). Using these centers, the clustering problem (7.2) is solved. Note that Algorithm 5.4 is utilized to solve both problems (7.2) and (7.3).

The following algorithm describes the IKMED in step by step.

Algorithm 7.4 Incremental k-medians algorithm (IKMED)

```
Input: Data set $A$ and the number of clusters $k$ to be computed.
Output: The $l$-partition of the set $A$ with $l = 1, \ldots, k$.

1: (Initialization) Compute the center $\boldsymbol{x}_1 \in \mathbb{R}^n$ of the set $A$. Set $l = 1$.
2: (Stopping criterion) Set $l = l + 1$. If $l > k$, then stop—the $k$-partition problem has been
   solved.
3: (Computation of a set of starting points for the auxiliary clustering problem) Apply
   Algorithm 7.2 to compute the set $\bar{A}_3$ of starting points for solving the auxiliary clustering
   problem (7.4).
4: (Computation of a set of starting points for the lth cluster center) Apply Algorithm 5.4 to
   solve the auxiliary clustering problem (7.4) starting from each point $\boldsymbol{y} \in \bar{A}_3$. This algorithm
   generates a set $\bar{A}_5$ of starting points for the $l$th cluster center.
5: (Computation of a set of cluster centers) For each $\bar{\boldsymbol{y}} \in \bar{A}_5$ apply Algorithm 5.4 for $k = l$ to
   solve the clustering problem (7.2) starting from the point $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}, \bar{\boldsymbol{y}})$ and find its solution
   $(\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_l)$. Denote by $\bar{A}_6$ a set of all such solutions.
6: (Computation of the best solution) Compute
       $f_l^{\min} = \min \Big\{ f_l(\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_l) : (\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_l) \in \bar{A}_6 \Big\},$
   and the collection of cluster centers $(\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_l)$ such that
       $f_l(\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_l) = f_l^{\min}.$
7: (Solution to the lth partition problem) Set $\boldsymbol{x}_j = \tilde{\boldsymbol{y}}_j,\ j = 1, \ldots, l$ as a solution to the $l$th
   partition problem and go to Step 2.
```

In Step 4 of Algorithm 7.4 one can apply the modified version of the k-medians Algorithm 5.4 to solve the auxiliary clustering problem and to find starting points for the lth cluster center. In this version, cluster centers $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}$ are fixed and the algorithm updates only the lth center. Therefore, we call it the *partial k-medians algorithm*. The description of this algorithm is given below.

We use the sets $\bar{S}_2$ and $\bar{B}_3(\boldsymbol{y}),\ \boldsymbol{y} \in \mathbb{R}^n$, defined in (7.8) and (7.9), respectively. Note that we employ the distance function d_1 in computing these sets.

Algorithm 7.5 Partial k-medians algorithm

Input: Data set A and the solution $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}) \in \mathbb{R}^{n(l-1)}$ to the $(l-1)$-partition problem.
Output: Solution $\bar{\boldsymbol{y}} \in \mathbb{R}^n$ to the auxiliary clustering problem (7.4).

```
1: (Initialization) Select a starting point $\boldsymbol{y}_1 \in \bar{S}_2$. Set $h = 1$.
2: Compute the set $\bar{B}_3(\boldsymbol{y}_h)$.
3: (Stopping criterion) If $\bar{B}_3(\boldsymbol{y}_h) = \bar{B}_3(\boldsymbol{y}_{h-1})$ for $h > 1$, then stop with the solution $\bar{\boldsymbol{y}} = \boldsymbol{y}_h$ to the auxiliary clustering problem.
4: Find a center $\bar{\boldsymbol{c}}$ of the set $\bar{B}_3(\boldsymbol{y}_h)$ by computing its coordinates as the medians of the corresponding attributes.
5: Set $\boldsymbol{y}_{h+1} = \bar{\boldsymbol{c}}$, $h = h + 1$ and go to Step 2.
```

Remark 7.3 The set $\bar{S}_2$ contains all data points $\boldsymbol{a} \in A$ which are not cluster centers and therefore, in Step 1 one can choose the point $\boldsymbol{y}_1$ among such data points. More specifically, we can choose $\boldsymbol{y}_1 \in A \setminus \bar{S}_1$ where the set $\bar{S}_1$ is given in (7.7). Furthermore, since for any $\boldsymbol{y} \in \bar{S}_2$ the set $\bar{B}_3(\boldsymbol{y})$ is not empty and the value of the auxiliary cluster function decreases at each iteration h the problem of finding the center of the sets $\bar{B}_3(\boldsymbol{y}_h),\ h \geq 1$ in Step 4 is well defined.

Note that the stopping criterion in Step 3 means that the algorithm terminates when no data point changes its cluster.

The most time-consuming steps in Algorithm 7.4 are Steps 3, 4, and 5. To reduce the computational effort required in these steps, we discuss three different approaches as follows:

1. *Reduction of the number of starting cluster centers.* As mentioned above, starting points for solving the auxiliary clustering problem (7.4) can be chosen from the set $A \setminus \bar{S}_1$. At the lth iteration ($l \geq 2$) of Algorithm 7.4, we can remove points that are close to cluster centers $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}$. For each cluster $A^q,\ 1 \leq q \leq l-1$, compute its average radius

   $$r_{av}^q = \frac{1}{|A^q|} \sum_{\boldsymbol{a} \in A^q} d_1(\boldsymbol{x}_q, \boldsymbol{a}),$$

   and define the subset $\hat{A}^q \subseteq A^q$ as

   $$\hat{A}^q = \big\{\boldsymbol{a} \in A^q \,:\, r_{av}^q \leq d_1(\boldsymbol{x}_q, \boldsymbol{a})\big\}.$$

   Note that if the cluster A^q is not empty, then the set $\hat{A}^q$ is also non-empty. Consider the following subset of the set A:

   $$\hat{A} = \bigcup_{q=1}^{l-1} \hat{A}^q.$$

Replacing the set $A \setminus \bar{S}_1$ by the set $\hat{A} \setminus \bar{S}_1$ allows us to reduce—in some cases significantly—the number of starting cluster centers and to remove those points which do not provide the sufficient decrease of the cluster function.

2. *Exclusion of some stationary points of the auxiliary clustering problem (7.4).* If any two stationary points from the set $\bar{A}_4$ are close to each other with respect to some predefined tolerance, then one of them is removed while another one is kept. In order to do so we define a tolerance $\varepsilon = \hat{f}_1/ml$, where $\hat{f}_1$ is the optimal value of the cluster function f_1. If $d_1(\boldsymbol{y}_1, \boldsymbol{y}_2) \leq \varepsilon$ for two points $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \bar{A}_4$, then the point with the lowest value of the auxiliary cluster function is kept in $\bar{A}_4$ and another point is removed.
3. *Use of the triangle inequality to reduce the number of distance calculations.* Since d_1 is the distance function it satisfies the triangle inequality. This can be used to reduce the number of distance function calculations of Algorithm 5.4 in solving both the clustering and the auxiliary clustering problems. First, we consider the auxiliary clustering problem (7.4). Assume that $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1})$ is the solution to the $(l-1)$-partition problem. Recall that the distance between the data point $\boldsymbol{a} \in A$ and its cluster center is denoted by

$$r_{l-1}^{\boldsymbol{a}} = \min_{j=1,\ldots,l-1} d_1(\boldsymbol{x}_j, \boldsymbol{a}).$$

Let $\bar{\boldsymbol{y}}$ be a current approximation to the solution of the problem (7.4). Compute distances $d_1(\bar{\boldsymbol{y}}, \boldsymbol{x}_j),\ j = 1, \ldots, l-1$. Assume that $\boldsymbol{a} \in A^j$ for some $j \in \{1, \ldots, l-1\}$. According to the triangle inequality we have

$$\begin{aligned} d_1(\bar{\boldsymbol{y}}, \boldsymbol{x}_j) &\leq d_1(\boldsymbol{a}, \bar{\boldsymbol{y}}) + d_1(\boldsymbol{a}, \boldsymbol{x}_j) = d_1(\boldsymbol{a}, \bar{\boldsymbol{y}}) + r_{l-1}^{\boldsymbol{a}}, \quad \text{or} \\ d_1(\boldsymbol{a}, \bar{\boldsymbol{y}}) &\geq d_1(\bar{\boldsymbol{y}}, \boldsymbol{x}_j) - r_{l-1}^{\boldsymbol{a}}. \end{aligned}$$

This means that if $d_1(\bar{\boldsymbol{y}}, \boldsymbol{x}_j) > 2r_{l-1}^{\boldsymbol{a}}$, then $d_1(\boldsymbol{a}, \bar{\boldsymbol{y}}) > r_{l-1}^{\boldsymbol{a}}$ and therefore, there is no need to calculate the distance $d_1(\boldsymbol{a}, \bar{\boldsymbol{y}})$ as the point $\boldsymbol{a}$ does not belong to the cluster with the center $\bar{\boldsymbol{y}}$.

Similar approach can be considered for the clustering problem (7.2). Let $(\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_l)$ be a current approximation to the solution of the lth partition problem. Compute distances $d_1(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_j)$ for $i, j = 1, \ldots, l$. Assume that for a given point $\boldsymbol{a} \in A$, the distances $d_1(\boldsymbol{a}, \bar{\boldsymbol{x}}_i),\ i = 1, \ldots, j$ have been calculated or estimated for some $j \in \{1, \ldots, l-1\}$. Let $\tilde{\boldsymbol{x}} \in \{\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_j\}$ be such that

$$d_1(\boldsymbol{a}, \tilde{\boldsymbol{x}}) = \min_{i=1,\ldots,j} d_1(\boldsymbol{a}, \bar{\boldsymbol{x}}_i).$$

According to the triangle inequality we have

$$d_1(\tilde{\boldsymbol{x}}, \bar{\boldsymbol{x}}_{j+1}) \le d_1(\boldsymbol{a}, \tilde{\boldsymbol{x}}) + d_1(\boldsymbol{a}, \bar{\boldsymbol{x}}_{j+1}), \quad \text{or}$$
$$d_1(\boldsymbol{a}, \bar{\boldsymbol{x}}_{j+1}) \ge d_1(\tilde{\boldsymbol{x}}, \bar{\boldsymbol{x}}_{j+1}) - d_1(\boldsymbol{a}, \tilde{\boldsymbol{x}}).$$

If $d_1(\tilde{\boldsymbol{x}}, \bar{\boldsymbol{x}}_{j+1}) > 2d_1(\boldsymbol{a}, \tilde{\boldsymbol{x}})$, then there is no need to calculate the distance $d_1(\boldsymbol{a}, \bar{\boldsymbol{x}}_{j+1})$ as the point $\boldsymbol{a}$ does not belong to the cluster A^{j+1} with the center $\bar{\boldsymbol{x}}_{j+1}$. The last approach allows us to significantly reduce the number of distance function evaluations as the number of clusters increases.