

# Chapter 9

## DC Optimization Based Clustering Algorithms



### 9.1 Introduction

This chapter presents the clustering algorithms based on the DC optimization approaches. In Chap. 4, the clustering problems are formulated using the DC representation of their objective functions. Using this representation we describe three different DC optimization algorithms.

For simplicity we use the following unconstrained DC programming problem to represent both the clustering and the auxiliary clustering problems (4.20) and (4.34):

$$\begin{cases} \text{minimize} & f(\mathbf{x}) = f_1(\mathbf{x}) - f_2(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathbb{R}^n, \end{cases} \quad (9.1)$$

where both  $f_1$  and  $f_2$  are finite valued convex functions on  $\mathbb{R}^n$ . As mentioned before, if the squared Euclidean norm is used to define the similarity measure, then the function  $f_1$  is smooth and the function  $f_2$  is, in general, nonsmooth. However, with other two similarity measures  $d_1$  and  $d_\infty$ , both functions are nonsmooth. In this chapter, we only consider the first case and present three different algorithms to solve the clustering problem (9.1).

We start with the incremental nonsmooth DC clustering algorithm [36]. This algorithm combines the MSINC-CLUST with the algorithm for finding inf-stationary points given in Fig. 3.7. The latter algorithm, in its turn, applies the NDCM presented in Fig. 3.8.

Then we present the DC diagonal bundle clustering algorithm [170]. Similar to the incremental DC clustering algorithm, the DC diagonal bundle clustering algorithm is a combination of the MSINC-CLUST and the NSO methods. However, here we apply the DCD-Bundle given in Fig. 3.6 instead of the NDCM.

Finally, we describe the incremental DCA for clustering [20]. The algorithm is a combination of the DCA (see Fig. 3.9) and the MSINC-CLUST.

## 9.2 Incremental Nonsmooth DC Clustering Algorithm

The *incremental nonsmooth DC clustering algorithm* (NDC-CLUST) is a combination of three different algorithms. The MSINC-CLUST is used to solve the clustering problem globally. At each iteration of this algorithm the algorithm for finding inf-stationary points is applied to solve both the clustering and the auxiliary clustering problems. In its turn, the later algorithm uses the NDCM to find Clarke stationary points of these problems. The flowchart of NDC-CLUST is given in Fig. 9.1.

Next, we present a detailed description of the NDC-CLUST. For a given point  $\mathbf{x} \in \mathbb{R}^n$  and a number  $\lambda > 0$ , consider the set

$$Q_1(\mathbf{x}, \lambda) = \text{conv} \{ \nabla f_1(\mathbf{x} + \lambda \mathbf{g}) : \mathbf{g} \in S_1 \},$$

where  $S_1$  is the sphere of the unit ball. It is obvious that the set  $Q_1(\mathbf{x}, \lambda)$  is convex and since the function  $f_1$  is smooth it is also compact for any  $\mathbf{x} \in \mathbb{R}^n$  and  $\lambda > 0$ .

Recall that a point  $\mathbf{x}^* \in \mathbb{R}^n$  is called  $(\lambda, \delta)$ -inf-stationary of the problem (9.1) if and only if

$$\partial f_2(\mathbf{x}^*) \subset Q_1(\mathbf{x}^*, \lambda) + B(\mathbf{0}; \delta),$$

and  $(\lambda, \delta)$ -stationary if there exists  $\xi_2 \in \partial f_2(\mathbf{x}^*)$  such that

$$\xi_2 \in Q_1(\mathbf{x}^*, \lambda) + B(\mathbf{0}; \delta).$$

If a point  $\mathbf{x} \in \mathbb{R}^n$  is not a  $(\lambda, \delta)$ -stationary point, then  $\|\xi_2 - \mathbf{z}\| \geq \delta$  for all  $\xi_2 \in \partial f_2(\mathbf{x})$  and  $\mathbf{z} \in Q_1(\mathbf{x}, \lambda)$ . Take any  $\xi_2 \in \partial f_2(\mathbf{x})$  and construct the set

$$\tilde{Q}(\mathbf{x}, \lambda, \xi_2) = Q_1(\mathbf{x}, \lambda) - \xi_2,$$

then we have

$$f(\mathbf{x} + \lambda \mathbf{u}) - f(\mathbf{x}) \leq \lambda \max_{\mathbf{z} \in \tilde{Q}(\mathbf{x}, \lambda, \xi_2)} \mathbf{z}^T \mathbf{u} \quad \text{for all } \mathbf{u} \in \mathbb{R}^n.$$

It is shown in Proposition 3.9 that if the point  $\mathbf{x}$  is not a  $(\lambda, \delta)$ -stationary, then the set  $\tilde{Q}(\mathbf{x}, \lambda, \xi_2)$  can be used to find a direction of sufficient decrease of the function  $f$  at  $\mathbf{x}$ . However, the computation of this set is not always possible. Next, we give a step by step algorithm which uses a finite number of elements from  $\tilde{Q}(\mathbf{x}, \lambda, \xi_2)$  to compute descent directions,  $(\lambda, \delta)$ -stationary points, and eventually Clarke stationary points of the problem (9.1). The flowchart and the more detailed

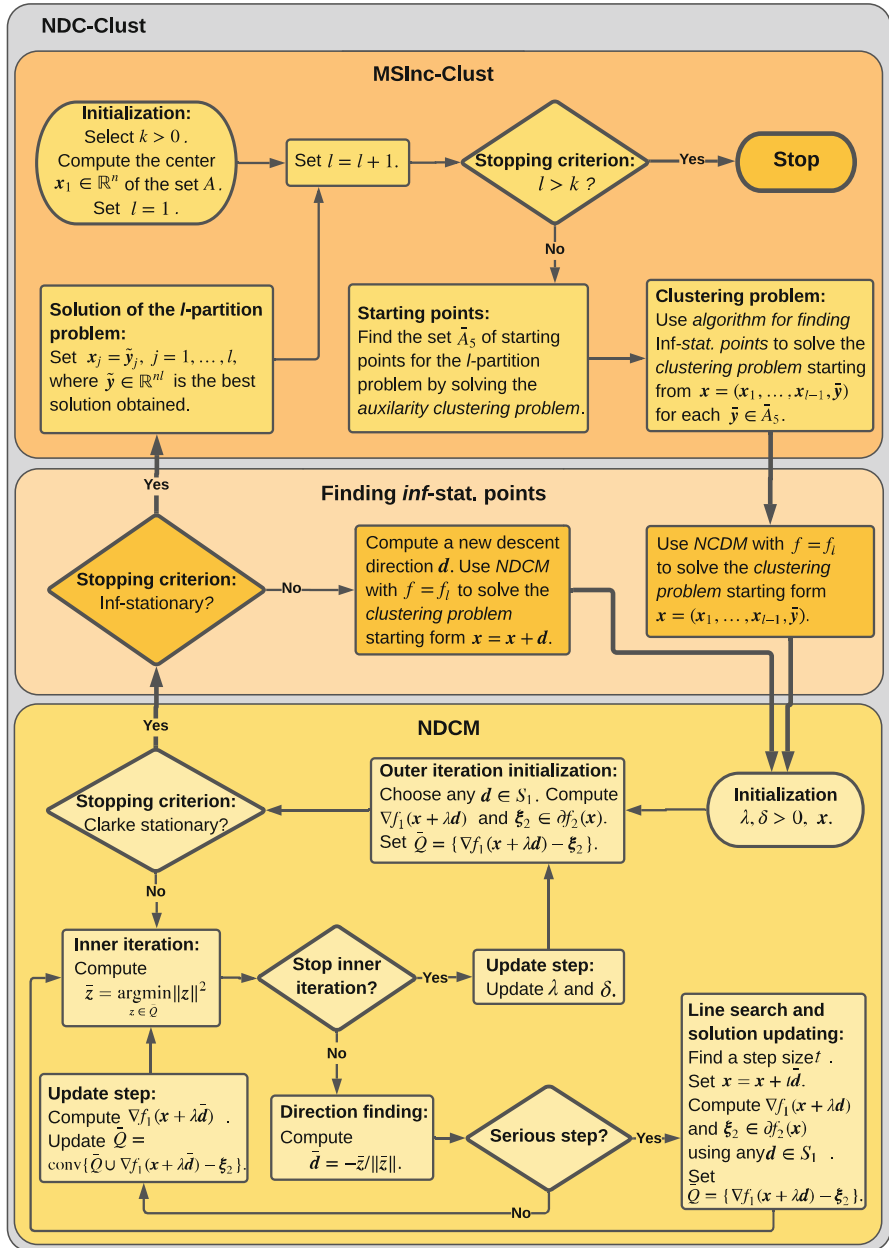


Fig. 9.1 Incremental nonsmooth DC clustering algorithm (NDC-CLUST)

description of this method (NDCM) are given in Sect. 3.6. Here, we use  $\mathbf{x}_1$  for the starting point;  $\varepsilon > 0$  for the stopping tolerance;  $\varepsilon_L$  and  $\varepsilon_R$  for line search parameters.

The convergence results for Algorithm 9.1 are given in Sect. 3.6. The next two propositions recall the most important results in light of the clustering problem.

**Proposition 9.1** *Algorithm 9.1 finds  $(\lambda, \delta)$ -stationary points of the clustering and the auxiliary clustering problems in at most  $h_{\max}$  iterations where*

$$h_{\max} = \left\lceil \frac{f(\mathbf{x}_1)}{\lambda \delta \varepsilon_R} \right\rceil.$$

*Proof* The proof follows from Proposition 3.10 and the fact that  $f^* = \inf\{f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n\} > 0$  for both the clustering and the auxiliary clustering problems.  $\square$

**Proposition 9.2** *Assume that  $\varepsilon = 0$ . Then all limit points of the sequence  $\{\mathbf{x}_h\}$  generated by Algorithm 9.1 are Clarke stationary points of the clustering or the auxiliary clustering problems.*

An algorithm for finding inf-stationary points of the problem (9.1) is presented next (see also Fig. 3.7). Assume that  $\mathbf{x}^*$  is a Clarke stationary point found by Algorithm 9.1. If the subdifferential  $\partial f_2(\mathbf{x}^*)$  is a singleton, then according to Proposition 3.7 the point is also an inf-stationary point.

**Algorithm 9.1** Nonsmooth DC algorithm

**Input:**  $\mathbf{x}_1 \in \mathbb{R}^n$ ,  $\varepsilon > 0$ ,  $\lambda_1 > 0$ ,  $\delta_1 > 0$ ,  $\varepsilon_L \in (0, 1)$  and  $\varepsilon_R \in (0, \varepsilon_L]$ .

**Output:** Approximate Clarke stationary point  $\mathbf{x}_h$ .

1: (*Outer iteration initialization*) Set  $h = 1$ .

2: (*Inner iteration initialization*) Set  $s = 1$  and  $\mathbf{x}_{h_s} = \mathbf{x}_h$ . Choose any  $\mathbf{g} \in S_1$  and compute  $\nabla f_1(\mathbf{x}_{h_s} + \lambda_h \mathbf{g})$  and  $\xi_{2,h_s} \in \partial f_2(\mathbf{x}_{h_s})$ . Set

$$\bar{Q}_h^s = \{ \nabla f_1(\mathbf{x}_{h_s} + \lambda_h \mathbf{g}) - \xi_{2,h_s} \}.$$

3: (*Stopping criterion*) If  $\lambda_h < \varepsilon$  and  $\delta_h < \varepsilon$ , then **stop** with  $\mathbf{x}_h$  as a final solution.

4: (*Minimum norm*) Compute

$$\mathbf{z}_{h_s} = \underset{\mathbf{z} \in \bar{Q}_h^s}{\operatorname{argmin}} \|\mathbf{z}\|^2.$$

5: (*Inner iteration termination*) If  $\|\mathbf{z}_{h_s}\| \leq \delta_h$ , then update  $\lambda_{h+1}$  and  $\delta_{h+1}$ . Set  $\mathbf{x}_{h+1} = \mathbf{x}_{h_s}$ ,  $h = h + 1$  and go to Step 2.

6: (*Search direction*) Compute the search direction

$$\mathbf{d}_{h_s} = -\frac{\mathbf{z}_{h_s}}{\|\mathbf{z}_{h_s}\|}.$$

7: If  $f(\mathbf{x}_{h_s} + \lambda_h \mathbf{d}_{h_s}) - f(\mathbf{x}_{h_s}) > -\varepsilon_L \lambda_h \|\mathbf{z}_{h_s}\|$ , then go to Step 9.

8: (*Serious step*) Construct  $\mathbf{x}_{h_{s+1}} = \mathbf{x}_{h_s} + t_{h_s} \mathbf{d}_{h_s}$ , where the step size  $t_{h_s}$  is computed as

$$t_{h_s} = \operatorname{argmax} \left\{ t \geq 0 : f(\mathbf{x}_{h_s} + t \mathbf{d}_{h_s}) - f(\mathbf{x}_{h_s}) \leq -\varepsilon_R t \|\mathbf{z}_{h_s}\| \right\}.$$

Choose any  $\mathbf{g} \in S_1$  and compute  $\nabla f_1(\mathbf{x}_{h_{s+1}} + \lambda_h \mathbf{g})$  and  $\xi_{2,h_{s+1}} \in \partial f_2(\mathbf{x}_{h_{s+1}})$ . Set

$$\bar{Q}_h^{s+1} = \left\{ \nabla f_1(\mathbf{x}_{h_{s+1}} + \lambda_h \mathbf{g}) - \xi_{2,h_{s+1}} \right\},$$

$s = s + 1$  and go to Step 4.

9: (*Null step*) Compute  $\nabla f_1(\mathbf{x}_{h_s} + \lambda_h \mathbf{d}_{h_s})$ . Update the set

$$\bar{Q}_h^{s+1} = \operatorname{conv} \left\{ \bar{Q}_h^s \cup \left\{ \nabla f_1(\mathbf{x}_{h_s} + \lambda_h \mathbf{d}_{h_s}) - \xi_{2,h_s} \right\} \right\}.$$

Set  $\mathbf{x}_{h_{s+1}} = \mathbf{x}_{h_s}$ ,  $s = s + 1$  and go to Step 4.

If the subdifferential  $\partial f_2(\mathbf{x}^*)$  is not a singleton, Corollary 3.3 implies that the point  $\mathbf{x}^*$  is not inf-stationary. Then according to Proposition 3.6 a descent direction from this point can be computed which in turn allows us to find a new starting point for Algorithm 9.1.

---

**Algorithm 9.2** Finding inf-stationary points of clustering problems
 

---

**Input:**  $\mathbf{x}_1 \in \mathbb{R}^n$ ,  $\varepsilon_A > 0$  and  $\varepsilon_T \in (0, 1/2]$ .

**Output:** Approximate inf-stationary point  $\mathbf{x}_j$ .

1: (*Initialization*) Set  $j = 1$ .

2: (*Clarke stationary point*) Apply Algorithm 9.1 starting from the point  $\mathbf{x}_j$  to find Clarke stationary point  $\mathbf{x}^*$  with the optimality tolerance  $\varepsilon_A$ .

3: (*Stopping criterion*) If

$$\partial f_2(\mathbf{x}^*) \subset \{\nabla f_1(\mathbf{x}^*)\} + B(\mathbf{0}; \varepsilon_A),$$

then **stop:**  $\mathbf{x}^*$  is an approximate inf-stationary point.

4: (*Descent direction*) Compute subgradients  $\xi_2^1, \xi_2^2 \in \partial f_2(\mathbf{x}^*)$  such that

$$r = \max_{i=1,2} \|\xi_2^i - \nabla f_1(\mathbf{x}^*)\| \geq \varepsilon_A,$$

and the direction  $\mathbf{u}_j = -\mathbf{v}/\|\mathbf{v}\|$  at  $\mathbf{x}^*$ , where

$$\mathbf{v} = \operatorname{argmax}_{i=1,2} \{\|\nabla f_1(\mathbf{x}^*) - \xi_2^i\|\}.$$

5: (*Step size*) Compute  $\mathbf{x}_{j+1} = \mathbf{x}^* + t_j \mathbf{u}_j$  where

$$t_j = \operatorname{argmax} \left\{ t \geq 0 : f(\mathbf{x}^* + t\mathbf{u}_j) - f(\mathbf{x}^*) \leq -\varepsilon_T t r \right\}.$$

Set  $j = j + 1$  and go to Step 2.

---

Note that if the subdifferential  $\partial f_2(\mathbf{x})$  is not singleton, then the two subgradients  $\xi_2^1, \xi_2^2 \in \partial f_2(\mathbf{x})$ , such that  $\xi_2^1 \neq \xi_2^2$  can be computed as described in Remarks 4.2 and 4.6. In addition, the following Lemmas show that the gradients of functions  $\bar{f}_{k1}$  and  $f_{k1}$ , given respectively in (4.33) and (4.19), satisfy Lipschitz condition.

**Lemma 9.1** *The gradient of the function  $\bar{f}_{k1}$  satisfies Lipschitz condition on  $\mathbb{R}^n$  with the constant  $L = 2$ .*

*Proof* Recall that the gradient of the function  $\bar{f}_{k1}$  at a point  $\mathbf{y} \in \mathbb{R}^n$  is

$$\nabla \bar{f}_{k1}(\mathbf{y}) = \frac{2}{m} \sum_{a \in A} (\mathbf{y} - \mathbf{a}).$$

Then for any  $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^n$  we get

$$\nabla \bar{f}_{k1}(\mathbf{y}_1) - \nabla \bar{f}_{k1}(\mathbf{y}_2) = 2(\mathbf{y}_1 - \mathbf{y}_2).$$

Therefore,

$$\|\nabla \bar{f}_{k1}(\mathbf{y}_1) - \nabla \bar{f}_{k1}(\mathbf{y}_2)\| = 2\|\mathbf{y}_1 - \mathbf{y}_2\|,$$

that is the gradient  $\nabla \bar{f}_{k1}$  satisfies the Lipschitz condition on  $\mathbb{R}^n$  with the constant  $L = 2$ .  $\square$

**Lemma 9.2** *The gradient of the function  $f_{k1}$  satisfies Lipschitz condition on  $\mathbb{R}^{nk}$  with the constant  $L = 2$ .*

*Proof* The proof is similar to that of Lemma 9.1.  $\square$

Considering clustering problems we can now get the following result.

**Proposition 9.3** *Algorithm 9.2 terminates after the finite number of iterations at an approximate inf-stationary point of the (auxiliary) clustering problem.*

*Proof* The proof follows directly from Proposition 3.8 and Lemmas 9.1 and 9.2.  $\square$

Now we are ready to give the NDC-CLUST for solving the problem (9.1). The NDC-CLUST first uses Algorithm 7.2 to generate a set of promising starting points for the auxiliary clustering problem. In addition, Algorithm 9.2 is utilized to solve both the clustering and the auxiliary clustering problems. This algorithm, in its turn, applies Algorithm 9.1 to find Clarke stationary points of the clustering problems. The NDC-CLUST is described in Algorithm 9.3.

---

**Algorithm 9.3** Incremental nonsmooth DC clustering algorithm (NDC-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 auxiliary clustering problem*) Apply Algorithm 7.2 to find the set  $\bar{A}_3 \subset \mathbb{R}^n$  of starting points for solving the auxiliary clustering problem (4.34).
- 4: (*Computation of a set of starting points for the  $l$ th cluster center*) Apply Algorithm 9.2 to solve the auxiliary clustering problem (4.34) starting from each point  $\mathbf{y} \in \bar{A}_3$ . 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  $\bar{\mathbf{y}} \in \bar{A}_5$  apply Algorithm 9.2 to solve the clustering problem (4.20) starting from the point  $(\mathbf{x}_1, \dots, \mathbf{x}_{l-1}, \bar{\mathbf{y}})$  and find a solution  $(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l)$ . Denote by  $\bar{A}_6$  a set of all such solutions.
- 6: (*Computation of the best solution*) Compute

$$f_l^{\min} = \min \left\{ f_l(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l) : (\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l) \in \bar{A}_6 \right\},$$

and the collection of cluster centers  $(\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_l)$  such that

$$f_l(\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_l) = f_l^{\min}.$$

- 7: (*Solution to the  $l$ th 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.
-

*Remark 9.1* Algorithm 9.3 can be used to solve clustering problems with the distance functions  $d_1$  and  $d_\infty$  if we apply the partial smoothing to the functions  $f_k$  and  $\bar{f}_k$ , described in Sects. 4.7.4 and 4.7.5, respectively (see [23]). More specifically, if we approximate the first component of the (auxiliary) cluster function by applying a smoothing technique then Algorithm 9.3 becomes applicable to solve clustering problems with the distance functions  $d_1$  and  $d_\infty$ .

### 9.3 DC Diagonal Bundle Clustering Algorithm

In this section, we describe the *DC diagonal bundle clustering algorithm* (DCDB-CLUST) for solving the problem (9.1) in large data sets [170]. The algorithm is a combination of three different algorithms. The MSINC-CLUST is used to solve the clustering problem globally. At each iteration of this algorithm a modified version of the algorithm for finding inf-stationary points (Algorithm 9.2) is applied to solve both the clustering and the auxiliary clustering problems. The later algorithm uses the DCD-BUNDLE to find Clarke stationary points of these problems. The flowchart of DCDB-CLUST is given in Fig. 9.2.

The DCD-BUNDLE is developed specifically to solve the clustering problems that are formulated as the nonsmooth DC optimization problem. The flowchart and more details of this method are given in Sect. 3.5. Here, we give the algorithm in its step by step form. We use  $\mathbf{x}_1$  for the starting point;  $\varepsilon_c > 0$  for the stopping



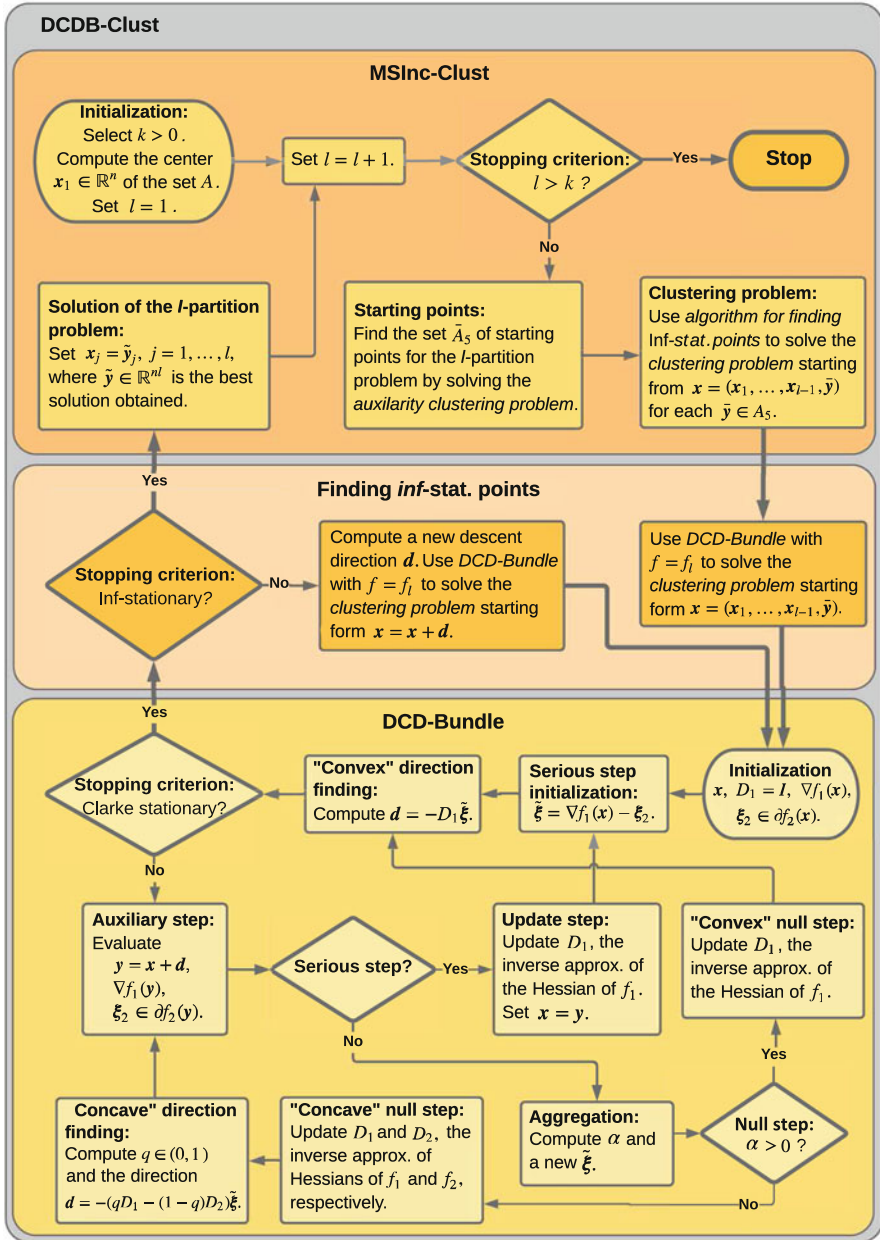


Fig. 9.2 DC diagonal bundle clustering algorithm (DCDB-CLUST)

---

**Algorithm 9.4** DC diagonal bundle algorithm
 

---

**Input:**  $\mathbf{x}_1 \in \mathbb{R}^n$ ,  $\varepsilon_c > 0$ ,  $\varepsilon_L \in (0, 1/2)$ ,  $\varepsilon_R \in (\varepsilon_L, 1)$ ,  $\hat{m}_c \geq 1$  and  $i_{\text{type}} \in \{0, 1\}$ .

**Output:** Clarke stationary point  $\mathbf{x}_h$ .

- 1: (*Initialization*) Set  $D_{1,1} = I$ . Compute  $f(\mathbf{x}_1)$ ,  $\nabla f_{1,1} = \nabla f_1(\mathbf{x}_1)$  and  $\xi_{2,1} \in \partial f_2(\mathbf{x}_1)$ . If  $i_{\text{type}} = 1$ , set  $\varepsilon = \varepsilon_c$ . Otherwise, set  $\varepsilon = 10^3 \varepsilon_c$ . Set  $h = 1$ .
- 2: (*Serious step initialization*) Set  $\tilde{\xi}_h = \xi_h = \nabla f_{1,h} - \xi_{2,h}$  and  $\tilde{\beta}_h = 0$ . Set  $m = h$ .
- 3: (*Convex direction*) Compute  $\mathbf{d}_h = -D_{1,h} \tilde{\xi}_h$ .
- 4: (*Stopping criterion*) Calculate  $w_h = \tilde{\xi}_h^T D_{1,h} \tilde{\xi}_h + 2\tilde{\beta}_h$ . If  $w_h < \varepsilon$ , then **stop** with  $\mathbf{x}_h$  as a final solution.
- 5: (*Auxiliary step*) Evaluate

$$\mathbf{y}_{h+1} = \mathbf{x}_h + \mathbf{d}_h, \quad \nabla f_{1,h+1} = \nabla f_1(\mathbf{y}_{h+1}) \quad \text{and} \quad \xi_{2,h+1} \in \partial f_2(\mathbf{y}_{h+1}).$$

Set  $s_h = \mathbf{d}_h$ ,  $\mathbf{u}_{1,h} = \nabla f_{1,h+1} - \nabla f_{1,m}$ ,  $\mathbf{u}_h = \xi_{2,h+1} - \xi_{2,m}$ , and add these values to the correction matrices  $S_h$ ,  $U_{1,h}$ , and  $U_{2,h}$  (delete the earliest values if  $|S_h| = |U_{1,h}| = |U_{2,h}| > \hat{m}_c$ ).

- 6: (*Serious step*) If

$$f(\mathbf{y}_{h+1}) - f(\mathbf{x}_h) \leq -\varepsilon_L w_h,$$

then compute  $D_{1,h+1}$  using  $S_h$  and  $U_{1,h}$ . Set  $\mathbf{x}_{h+1} = \mathbf{y}_{h+1}$ ,  $f(\mathbf{x}_{h+1}) = f(\mathbf{y}_{h+1})$  and go to Step 2.

- 7: (*Aggregation*) Compute

$$\alpha_{h+1} = f(\mathbf{x}_h) - f(\mathbf{y}_{h+1}) + (\nabla f_{1,h+1} - \xi_{2,h+1})^T \mathbf{d}_h,$$

and  $t \in (0, 1]$  such that  $\xi'_{h+1} \in \partial f(\mathbf{x}_h + t\mathbf{d}_h)$  satisfies the condition

$$-\beta_{h+1} + (\xi'_{h+1})^T \mathbf{d}_h \geq -\varepsilon_R w_h,$$

with  $\beta_{h+1}$  given in (3.19). Determine multipliers  $\lambda_i^k \geq 0$  for all  $i \in \{1, 2, 3\}$ ,  $\sum_{i=1}^3 \lambda_i^h = 1$  that minimize the function

$$\begin{aligned} \varphi(\lambda_1, \lambda_2, \lambda_3) &= (\lambda_1 \xi_m + \lambda_2 \xi'_{h+1} + \lambda_3 \tilde{\xi}_h)^T D_{1,h} (\lambda_1 \xi_m + \lambda_2 \xi'_{h+1} + \lambda_3 \tilde{\xi}_h) \\ &\quad + 2(\lambda_2 \beta_{h+1} + \lambda_3 \tilde{\beta}_h). \end{aligned}$$

Set  $\tilde{\xi}'_{h+1} = \lambda_1^h \xi_m + \lambda_2^h \xi'_{h+1} + \lambda_3^h \tilde{\xi}_h$  and  $\tilde{\beta}_{h+1} = \lambda_2^h \beta_{h+1} + \lambda_3^h \tilde{\beta}_h$ .

- 8: (*Null step*) If  $m = h$ , then compute  $D_{1,h+1}$  using  $S_h$  and  $U_{1,h}$ . Otherwise, set  $D_{1,h+1} = D_{1,h}$ . Two cases can occur.
  - (i) (*Convex Null Step*) If  $\alpha_{h+1} \geq 0$ , then set  $\mathbf{x}_{h+1} = \mathbf{x}_h$ ,  $h = h + 1$  and go to Step 3.
  - (ii) (*Concave Null Step*) If  $\alpha_{h+1} < 0$ , then compute  $D_{2,h+1}$  using  $S_h$  and  $U_{2,h}$ . Set  $\mathbf{x}_{h+1} = \mathbf{x}_h$ ,  $h = h + 1$ .
- 9: (*Concave direction*) Compute the smallest  $q \in (0, 1)$  such that the matrix  $qD_{1,h} - (1-q)D_{2,h}$  remains positive semidefinite. Compute

$$\mathbf{d}_h = -(qD_{1,h} - (1-q)D_{2,h}) \tilde{\xi}_h$$

and go to Step 4.

---

tolerance;  $\varepsilon_L$  and  $\varepsilon_R$  for line search parameters;  $\gamma$  for the distance measure parameter;  $\hat{m}_c$  for the maximum number of stored correction vectors used to form diagonal updates. We also use  $i_{\text{type}}$  to show the type of the problem, that is:

- $i_{\text{type}} = 0$ : the auxiliary clustering problem (7.4);
- $i_{\text{type}} = 1$ : the clustering problem (7.2).

The convergence properties of the DCD-BUNDLE are studied in Sect. 3.5. Here, we recall the most important results for clustering problems. Note that Assumptions 3.5–3.6 are trivially satisfied for both the cluster and the auxiliary cluster functions.

**Proposition 9.4** *Assume  $\varepsilon_c = 0$ . If Algorithm 9.4 terminates at the  $h$ th iteration, then the point  $\mathbf{x}_h$  is a Clarke stationary point of the (auxiliary) clustering problem.*

**Proposition 9.5** *Assume  $\varepsilon_c = 0$ . Every accumulation point of the sequence  $\{\mathbf{x}_h\}$  generated by Algorithm 9.4 is a Clarke stationary of the (auxiliary) clustering problem.*

If the function  $f_2$  in the problem (9.1) is smooth, then the point found by Algorithm 9.4 is also inf-stationary. Otherwise, a slight modification of Algorithm 9.2 is applied to find an inf-stationary point of the problem. This modification is given in Algorithm 9.5.

**Algorithm 9.5** Finding inf-stationary points of clustering problems

**Input:**  $\mathbf{x}_1 \in \mathbb{R}^n$ ,  $\varepsilon_A > 0$  and  $\varepsilon_T \in (0, 1/2]$ .

**Output:** Approximate inf-stationary point  $\mathbf{x}^*$ .

- 1: (*Initialization*) Set  $j = 1$ .
- 2: (*Clarke stationary point*) Apply Algorithm 9.4 starting from the point  $\mathbf{x}_j$  to find the Clarke stationary point  $\mathbf{x}^*$  with the optimality tolerance  $\varepsilon_A$ .
- 3: (*Stopping criterion*) If

$$\partial f_2(\mathbf{x}^*) \subset \{\nabla f_1(\mathbf{x}^*)\} + B(\mathbf{0}; \varepsilon_A),$$

then **stop**:  $\mathbf{x}^*$  is an approximate inf-stationary point.

- 4: (*Descent direction*) Compute subgradients  $\xi_2^1, \xi_2^2 \in \partial f_2(\mathbf{x}^*)$  such that

$$r = \max_{i=1,2} \|\xi_2^i - \nabla f_1(\mathbf{x}^*)\| \geq \varepsilon_A,$$

and the direction  $\mathbf{u}_j = -\mathbf{v}/\|\mathbf{v}\|$  at  $\mathbf{x}^*$ , where

$$\mathbf{v} = \operatorname{argmax}_{i=1,2} \|\nabla f_1(\mathbf{x}^*) - \xi_2^i\|.$$

- 5: (*Step size*) Compute  $\mathbf{x}_{j+1} = \mathbf{x}^* + t_j \mathbf{u}_j$  where

$$t_j = \operatorname{argmax} \left\{ t \geq 0 : f(\mathbf{x}^* + t\mathbf{u}_j) - f(\mathbf{x}^*) \leq -\varepsilon_T t r \right\}.$$

Set  $j = j + 1$  and go to Step 2.

If the subdifferential  $\partial f_2(\mathbf{x})$  is not a singleton, then we can compute two different subgradients  $\xi_2^1, \xi_2^2 \in \partial f_2(\mathbf{x})$  in Step 4 of Algorithm 9.5 (see Remarks 4.2 and 4.6). In addition, in Lemmas 9.1 and 9.2, we proved that the gradients of functions  $\tilde{f}_{k1}$  and  $f_{k1}$  (see (4.20) and (4.34)) satisfy the Lipschitz condition. Then we get the following convergence result for clustering problems.

**Proposition 9.6** *Algorithm 9.5 terminates after finite number of iterations at an approximate inf-stationary point of the (auxiliary) clustering problem.*

*Proof* The proof follows directly from Proposition 3.8 and Lemmas 9.1 and 9.2.  $\square$

Next, we give the step by step description of the DCDB-CLUST.

---

**Algorithm 9.6** DC diagonal bundle clustering algorithm (DCDB-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 auxiliary clustering problem*) Apply Algorithm 7.2 to find the set  $\bar{A}_3 \subset \mathbb{R}^n$  of starting points for the auxiliary clustering problem (4.34).
- 4: (*Computation of a set of starting points for the clustering problem*) For each  $\mathbf{y} \in \bar{A}_3$  apply Algorithm 9.5 to solve the auxiliary clustering problem (4.34) and find  $\bar{A}_5$ , a set of starting points for the  $l$ th cluster center in the  $l$ th clustering problem (4.20).
- 5: (*Computation of a set of cluster centers*) For each  $\bar{\mathbf{y}} \in \bar{A}_5$  apply Algorithm 9.5 to solve the clustering problem (4.20) starting from the point  $(\mathbf{x}_1, \dots, \mathbf{x}_{l-1}, \bar{\mathbf{y}})$  and find a solution  $(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l)$ . Denote by  $\bar{A}_6$  a set of all such solutions.
- 6: (*Computation of the best solution*) Compute

$$f_l^{\min} = \min \left\{ f_l(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l) : (\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l) \in \bar{A}_6 \right\},$$

and the collection of cluster centers  $(\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_l)$  such that

$$f_l(\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_l) = f_l^{\min}.$$

- 7: (*Solution to the  $l$ th 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.
- 

*Remark 9.2* Similar to Algorithm 9.3, Algorithm 9.6 can be applied to solve clustering problems with the distance functions  $d_1$  and  $d_\infty$  if we apply the partial smoothing to the cluster function  $f_k$  and the auxiliary cluster function  $\tilde{f}_k$ .

## 9.4 Incremental DCA for Clustering

In this section, we describe an *incremental DCA for clustering* (IDCA-CLUST) to solve the clustering problem (9.1) [20]. The IDCA-CLUST is based on the MSINC-CLUST and the DCA, where the latter algorithm is utilized at each iteration of the MSINC-CLUST to solve the clustering and the auxiliary clustering problems. Figure 9.3 illustrates the flowchart of the IDCA-CLUST.

First, we recall the DCA for solving the unconstrained DC programming problem (9.1) when the first DC component  $f_1$  is continuously differentiable.

---

### Algorithm 9.7 DC algorithm

---

**Input:** Starting point  $\mathbf{x}_1 \in \mathbb{R}^n$ .

**Output:** Critical point  $\mathbf{x}_h$ .

- 1: (*Initialization*) Set  $h = 1$ .
- 2: Compute  $\xi_{2,h} \in \partial f_2(\mathbf{x}_h)$ .
- 3: (*Stopping criterion*) If  $\xi_{2,h} = \nabla f_1(\mathbf{x}_h)$ , then **stop**.
- 4: Find the solution  $\mathbf{x}_{h+1}$  to the convex optimization problem

$$\begin{cases} \text{minimize} & f_1(\mathbf{x}) - \xi_{2,h}^T (\mathbf{x} - \mathbf{x}_h) \\ \text{subject to} & \mathbf{x} \in \mathbb{R}^n. \end{cases} \quad (9.2)$$

- 5: Set  $h = h + 1$  and go to Step 2.
- 

Next, we explain how this algorithm can be applied to solve the clustering and the auxiliary clustering problems (9.1). We start with the clustering problem. Let  $\mathbf{x}_h = (\mathbf{x}_{h,1}, \dots, \mathbf{x}_{h,k}) \in \mathbb{R}^{nk}$  be a vector of cluster centers at the iteration  $h$  and  $A^1, \dots, A^k$  be the cluster partition of the data set  $A$  provided by these centers.

We discussed the subdifferentials of the functions  $f_1$  and  $f_2$  in Sect. 4.4. Here, we recall them when the similarity measure  $d_2$  is used in these functions. In this case, the function  $f_1$  is continuously differentiable and we have

$$\nabla f_{k1}(\mathbf{x}) = 2(\mathbf{x} - \tilde{\mathbf{a}}), \quad \mathbf{x} \in \mathbb{R}^{nk},$$

where  $\tilde{\mathbf{a}} = (\bar{\mathbf{a}}, \dots, \bar{\mathbf{a}})$  and  $\bar{\mathbf{a}} = \frac{1}{m} \sum_{i=1}^m \mathbf{a}_i$ .

For the subdifferential of the function  $f_2$ , recall the function  $\varphi_{\mathbf{a}}(\mathbf{x})$  and the set  $\tilde{\mathcal{R}}_{\mathbf{a}}(\mathbf{x})$ ,  $\mathbf{x} \in \mathbb{R}^{nk}$ , defined in (4.22) and (4.23), respectively:

$$\varphi_{\mathbf{a}}(\mathbf{x}) = \max_{j=1, \dots, k} \sum_{s=1, s \neq j}^k d_2(\mathbf{x}_s, \mathbf{a}),$$

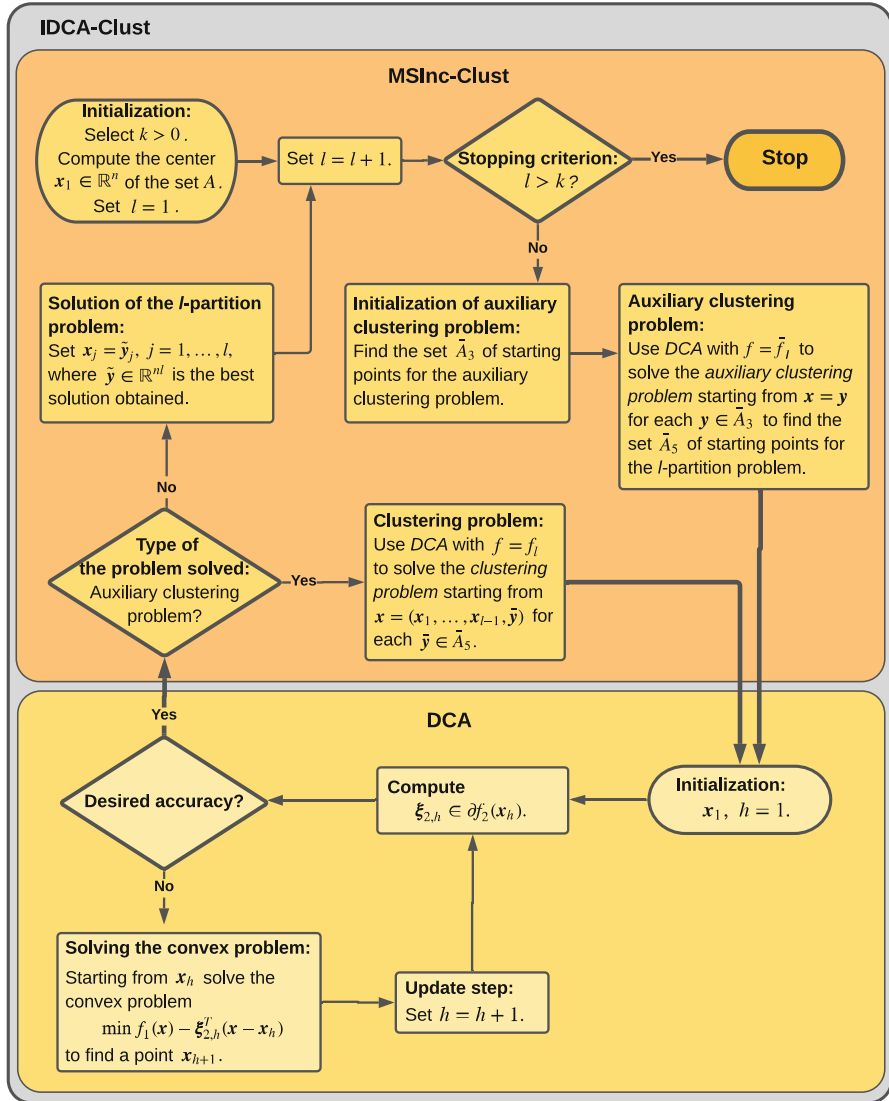


Fig. 9.3 Incremental DCA for clustering (IDCA-CLUST)

and

$$\tilde{\mathcal{R}}_a(\mathbf{x}) = \left\{ j \in \{1, \dots, k\} : \sum_{s=1, s \neq j}^k d_2(\mathbf{x}_s, \mathbf{a}) = \varphi_a(\mathbf{x}) \right\}.$$

Then we have

$$\partial\varphi_a(\mathbf{x}) = \text{conv} \left\{ 2(\mathbf{x}_1 - \mathbf{a}, \dots, \mathbf{x}_{j-1} - \mathbf{a}, \mathbf{0}, \mathbf{x}_{j+1} - \mathbf{a}, \dots, \mathbf{x}_k - \mathbf{a}), \right. \\ \left. j \in \tilde{\mathcal{R}}_a(\mathbf{x}) \right\},$$

and

$$\partial f_{k2}(\mathbf{x}) = \frac{1}{m} \sum_{\mathbf{a} \in A} \partial\varphi_a(\mathbf{x}).$$

Applying these formulas for subdifferentials, the subgradient  $\xi_{2,h} \in \partial f_2(\mathbf{x}_h)$  in Step 2 of Algorithm 9.7 is

$$\xi_{2,h} = \frac{2}{m} \left( \sum_{\mathbf{a} \in A \setminus A^1} (\mathbf{x}_{h,1} - \mathbf{a}), \dots, \sum_{\mathbf{a} \in A \setminus A^k} (\mathbf{x}_{h,k} - \mathbf{a}) \right) \\ = \frac{2}{m} \left( (m - |A^1|)\mathbf{x}_{h,1} - (m\bar{\mathbf{a}} - |A^1|\bar{\mathbf{a}}_1), \dots, \right. \\ \left. (m - |A^k|)\mathbf{x}_{h,k} - (m\bar{\mathbf{a}} - |A^k|\bar{\mathbf{a}}_k) \right),$$

where  $\bar{\mathbf{a}}_l$  is the center of the cluster  $A^l$ ,  $l = 1, \dots, k$  and  $\bar{\mathbf{a}}$  is the center of the whole set  $A$ . In addition, the solution  $\mathbf{x}_{h+1} = (\mathbf{x}_{h+1,1}, \dots, \mathbf{x}_{h+1,k})$  to the problem (9.2) in Step 4 of Algorithm 9.7 is

$$\mathbf{x}_{h+1,t} = \left( 1 - \frac{|A^t|}{m} \right) \mathbf{x}_{h,t} + \frac{|A^t|}{m} \bar{\mathbf{a}}_t, \quad t = 1, \dots, k,$$

and the stopping criterion in Step 3 of this algorithm can be given as

$$\mathbf{x}_{h,t} = \left( 1 - \frac{|A^t|}{m} \right) \mathbf{x}_{h,t} + \frac{|A^t|}{m} \bar{\mathbf{a}}_t, \quad t = 1, \dots, k.$$

In order to apply Algorithm 9.7 for solving the auxiliary clustering problem, recall the sets  $B_i(\mathbf{y})$ ,  $i = 1, 2, 3$ , defined in (4.30) for  $p = 2$  and  $\mathbf{y} = \mathbf{x}_h \in \mathbb{R}^n$ :

$$B_1(\mathbf{x}_h) = \{ \mathbf{a} \in A : r_{l-1}^a < d_2(\mathbf{x}_h, \mathbf{a}) \}, \\ B_2(\mathbf{x}_h) = \{ \mathbf{a} \in A : r_{l-1}^a = d_2(\mathbf{x}_h, \mathbf{a}) \}, \quad \text{and}$$

$$B_3(\mathbf{x}_h) = \{\mathbf{a} \in A : r_{l-1}^{\mathbf{a}} > d_2(\mathbf{x}_h, \mathbf{a})\}.$$

Then the subgradient  $\xi_{2,h} \in \partial f_2(\mathbf{x}_h)$  in Step 2 of Algorithm 9.7 is computed as

$$\xi_{2,h} = \frac{2}{m} \sum_{\mathbf{a} \in B_1(\mathbf{x}_h)} (\mathbf{x}_h - \mathbf{a}), \quad \mathbf{x}_h \in \mathbb{R}^n.$$

Furthermore, the solution  $\mathbf{x}_{h+1}$  to the problem (9.2) in Step 4 is

$$\mathbf{x}_{h+1} = \frac{1}{m} \left( |B_1(\mathbf{x}_h)| \mathbf{x}_h + \sum_{\mathbf{a} \in B_2(\mathbf{x}_h) \cup B_3(\mathbf{x}_h)} \mathbf{a} \right).$$

Finally, the stopping criterion in Step 3 of Algorithm 9.7 can be given by

$$\sum_{\mathbf{a} \in B_2(\mathbf{x}_h) \cup B_3(\mathbf{x}_h)} (\mathbf{x}_h - \mathbf{a}) = 0.$$

These results demonstrate that there is no need to apply any optimization algorithm to solve the problem (9.2) for both the DC clustering and the DC auxiliary clustering problems. In both cases solutions can be expressed explicitly.

**Proposition 9.7** *All accumulation points of the sequence  $\{\mathbf{x}_h\}$  generated by Algorithm 9.7 are Clarke stationary points of the problem (9.1) when  $d_2$  is used as a similarity measure.*

*Proof* Since the function  $f_1$  in the problem (9.1) with the similarity measure  $d_2$  is smooth the sets of critical points and Clarke stationary points of this problem coincide (see Theorem 2.27 and Fig. 2.9).  $\square$

Now, we are ready to design an IDCA-CLUST. This algorithm is based on the MSINC-CLUST and the DCA. The IDCA-CLUST applies the MSINC-CLUST for solving the clustering problem globally and the DCA is utilized at each iteration of the MSINC-CLUST to solve both the clustering and the auxiliary clustering problems. The step by step description of the IDCA-CLUST is given in Algorithm 9.8.

*Remark 9.3* Similar to Algorithms 9.3 and 9.6, we can apply Algorithm 9.2 in Steps 4 and 5 of Algorithm 9.8. Then we obtain inf-stationary points of both the clustering and the auxiliary clustering problems.



---

**Algorithm 9.8** Incremental DCA for clustering (IDCA-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 auxiliary clustering problem*) Apply Algorithm 7.2 to find the set  $\bar{A}_3$  of starting points for solving the auxiliary clustering problem (4.34).
- 4: (*Computation of a set of starting points for the  $l$ th cluster center*) Apply Algorithm 9.7 to solve the auxiliary clustering problem (4.34) starting from each point  $\mathbf{y} \in \bar{A}_3$ . 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  $\bar{\mathbf{y}} \in \bar{A}_5$  apply Algorithm 9.7 to solve the clustering problem (4.20) starting from the point  $(\mathbf{x}_1, \dots, \mathbf{x}_{l-1}, \bar{\mathbf{y}})$  and find a solution  $(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l)$ . Denote by  $\bar{A}_6$  a set of all such solutions.
- 6: (*Computation of the best solution*) Compute

$$f_l^{\min} = \min \left\{ f_l(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l) : (\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_l) \in \bar{A}_6 \right\},$$

and the collection of cluster centers  $(\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_l)$  such that

$$f_l(\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_l) = f_l^{\min}.$$

- 7: (*Solution to the  $l$ th 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.
-