frontiers | Frontiers in Applied Mathematics and Statistics

TYPE Original Research
PUBLISHED 12 September 2024
DOI 10.3389/fams.2024.1445390

OPEN ACCESS

EDITED BY
Lixin Shen,
Syracuse University, United States

REVIEWED BY
Jianqing Jia,
Syracuse University, United States
Rongrong Lin,
Guangdong University of Technology, China

*CORRESPONDENCE
Adugna Fita Gabissa
✉ adugna.fita@astu.edu.et

RECEIVED 17 June 2024
ACCEPTED 09 August 2024
PUBLISHED 12 September 2024

CITATION
Gabissa AF and Obsu LL (2024) A DC programming to two-level hierarchical clustering with ℓ_1 norm.
Front. Appl. Math. Stat. 10:1445390.
doi: 10.3389/fams.2024.1445390

A DC programming to two-level hierarchical clustering with ℓ_1 norm

Adugna Fita Gabissa* and Legesse Lemecha Obsu

Department of Applied Mathematics, Adama Science and Technology University, Adama, Ethiopia

The main challenge in solving clustering problems using mathematical optimization techniques is the non-smoothness of the distance measure used. To overcome this challenge, we used Nesterov's smoothing technique to find a smooth approximation of the ℓ_1 norm. In this study, we consider a bi-level hierarchical clustering problem where the similarity distance measure is induced from the ℓ_1 norm. As a result, we are able to design algorithms that provide optimal cluster centers and headquarter (HQ) locations that minimize the total cost, as evidenced by the obtained numerical results.

KEYWORDS

clustering, DC programming, bi-level hierarchical, headquarter, smoothing

1 Introduction

Clustering, a widely studied field with applications across various scientific and engineering domains, often grapples with non-smooth and non-convex problems that defy traditional gradient descent algorithms. The discrete and combinatorial nature of clustering adds another layer of complexity, making optimality challenging to attain.

The synergy of Nesterov's smoothing technique [16], DC programming, and the difference of convex algorithm (DCA) [10] has created a fertile ground for investigating into non-convex and non-smooth optimization problems. The efficacy of the DC algorithm in addressing non-convex clustering problems has been well-established in previous studies [1, 5, 14, 17, 22] and cited references. Notable among these is the exploration of a DC optimization approach for constrained clustering with ℓ_1 norm [6], tackling problems such as the minimum sum of squares clustering [2], bi-level hierarchical clustering [8], and multicast network design [13]. Recent studies have extended DC algorithms to solve multifacility location problems [4] and addressed similar issues using alternative approaches [21].

While previous methods often resorted to meta-heuristic algorithms, which are challenging to analyze for optimality, recent advancements have seen a shift toward more robust techniques. In 2003, Jia et al. [9] introduced three models of hierarchical clustering based on the Euclidean norm and employed the derivative-free method developed in [3] to solve the problem in $\mathbb{R}^2$. In [21], DCA which was developed in [19, 20] was utilized by replacing ℓ_2 norm by squared ℓ_2 norm and applied to higher dimensional problems. However, the need for further enhancements led to the incorporation of new way in Nesterov's smoothing techniques in [8, 13] to overcome certain limitations identified in [9].

In real-world scenarios, the ℓ_1 distance measure frequently provides a more accurate reflection of ground realities than the Euclidean distance. This study extends the investigation of the bi-level hierarchical clustering model proposed in [8, 13] by modifying the objective function and constraints using the ℓ_1 norm. Employing Nesterov's partial smoothing techniques and a suitable DC decomposition tailored for the ℓ_1 norm, we leverage the DC Algorithm (DCA). In addition, constraints are introduced to ensure that cluster centers and the headquarters lie on actual nodes in the datasets. To limit the search

space, the headquarter is strategically placed in the region average to the cluster centers that minimize the overall distance of the network.

The study is organized as follows: Section 2 introduces the basic tools of convex analysis applied to DC functions and DCA. Sections 3 and subsequent subsections delve into the formulation and exploration of bi-level hierarchical clustering problems, along with the development of DCA algorithms that address the model using Nesterov's smoothing technique. Section 4 showcases numerical simulation results with artificial data, and concluding remarks are presented in Section 5.

2 Fundamentals of convex analysis

In this section, we will introduce fundamental results and definitions from convex analysis, crucial for understanding the subsequent discussions in this study. For in-depth technical proofs and additional readings, we recommend referring to [11, 12].

Definition 1. An extended real-valued function $f:\mathbb{R}^n \to (-\infty,\infty]$ is called a DC function, if it can be represented as a difference of two convex functions g and h.

Moreover, the optimization problem

$$\text{minimize } f(x) := g(x) - h(x); \quad x \in \mathbb{R}^n \tag{1}$$

referred to as a DC optimization problem, and it can be addressed using the *difference of convex algorithm* introduced by Tao and An[19, 20] as follows.

```
Input: x_0 ∈ R^n, N ∈ N;
while stopping is not reached do
    for k = 1, ..., N do
        Find y_k ∈ ∂h(x_{k-1});
        Find x_k ∈ ∂g*(y_k);
    end for;
end while;
Return x_N.
```

Algorithm 1. DCA algorithm 1.

The function g^* referred in the DCA is the *Fenchel Conjugate* of g, and it is defined as in [18]

$$g^*(y) = \sup\{\langle y,x\rangle - g(x) \mid x \in \mathbb{R}^n\}, y \in \mathbb{R}^n, \tag{2}$$

and it is always convex regardless of whether g is convex or not.

Theorem 1. [18] Let $g:\mathbb{R}^n \to (-\infty,\infty]$ be a proper extended real-valued function, for $x,y \in \mathbb{R}^n$. Then, $x \in \partial g^*(y)$ if and only if $y \in \partial g(x)$.

Definition 2. [12] A vector $v \in \mathbb{R}^n$ is a *sub-gradient* of a convex function $f:\mathbb{R}^n \to (-\infty,\infty]$, at $\bar{x} \in dom(f)$, if it satisfies the inequality

$$f(x) \geq f(\bar{x}) + \langle v, x-\bar{x}\rangle \quad \text{for all} \quad x \in \mathbb{R}^n.$$

The set of all sub-gradients of f at $\bar{x}$, denoted as $\partial f(\bar{x})$, is known as the sub-differential of f at $\bar{x}$, that is,

$$\partial f(\bar{x}) = \{v \in \mathbb{R}^n \mid f(x) \geq f(\bar{x}) + \langle v, x-\bar{x}\rangle \quad \text{for all} \quad x \in \mathbb{R}^n\}. \tag{3}$$

Theorem 2. Let $f_i:\mathbb{R}^n \to (-\infty,\infty]$ be a proper and extended real-valued convex function on $\mathbb{R}^n$, where $i = 1,2,\ldots,m$ and $\bigcap_{i=1}^m rint(dom(f_i)) \neq \emptyset$ [12]. Then for all $\bar{x} \in \bigcap_{i=1}^m dom(f_i)$,

$$\partial\left(\sum_{i=1}^m f_i(\bar{x})\right) = \sum_{i=1}^m \partial f_i(\bar{x}).$$

2.1 The max, min, and convergence of the DCA

The *maximum function* is defined as the point-wise maximum of convex functions. For $i = 1,2,3,\ldots,m$, let the functions $f_i:\mathbb{R}^n \to \mathbb{R}$ be closed and convex. Then, the maximum function

$$f(x) := \max_{i=1,\ldots,m} f_i(x) = \max\left\{f_1(x), f_2(x), \ldots, f_m(x)\right\},$$

is also closed and convex. On the other hand, the *minimum function* $f(x)$, defined by

$$f(x) := \min_{i=1,\ldots,m} f_i(x) = \min\left\{f_1(x), f_2(x), \ldots, f_m(x)\right\}$$

may not be convex. However, it can always be represented as a difference of two convex functions as follows:

$$\min\left\{f_1(x), f_2(x), \ldots, f_m(x)\right\} = \sum_{i=1}^m f_i(x) - \max_{t=1,\ldots,m} \sum_{i=1, i\neq t}^m f_i(x). \tag{4}$$

Lemma 3. [12] Let functions $f_i(x)$, $i = 1 \ldots m$ be closed and convex. Then, the maximum function

$$f(x) = \max\left\{f_1(x), \ldots, f_m(x)\right\}$$

is also closed and convex. Moreover, for any $x \in int(domf) = \cap_{i=1}^m int(domf_i)$, we have

$$\partial f(x) = Conv\left\{\partial f_i(x) \mid i \in I(x)\right\},$$

where $I(x) = \left\{i : f_i(x) = f(x)\right\}$.

Definition 3. [14] A function $f:\mathbb{R}^n \to \mathbb{R}$ is ρ-strongly convex if there exists $\rho > 0$ such that the function

$$g(x) := f(x) - \frac{\rho}{2}\|x\|^2$$

is convex. In particular, if f is strongly convex, then f is also strictly convex, in the sense that $f(\lambda x_1 + (1-\lambda)x_2) < \lambda f(x_1) + (1-\lambda)f(x_2)$ for all $\lambda \in (0,1)$.

Theorem 4. [14, 20] Let f be as defined in problem (1), and let $\{x_k\}$ be a sequence generated by the DCA Algorithm 1. Suppose that g and h are ρ_1 and ρ_2 strongly convex, respectively. Then, at every iteration number k of the DCA, we have

$$f(x_{k+1}) \leq f(x_k) - \frac{\rho_1 + \rho_2}{2}\|x_{k+1} - x_k\|^2. \tag{5}$$

Moreover, if f is bounded from below and if $\{x_k\}$ is bounded, then all sub-sequential limits of $\{x_k\}$ converge to a stationary point of f.

2.2 Nesterov's smoothing approximation of the ℓ_1 Norm

Definition 4. [12, 14] Let F be a non-empty closed subset of $\mathbb{R}^n$ and let $x \in \mathbb{R}^n$.

1. Define the distance between x and set F by

$$d_F(x) = \inf\{\|x - w\| \mid w \in F\}.$$

2. The set of all Euclidean projection from x to F is defined by

$$P(x; F) = \{w \in F \mid d_F(x) = \|x - w\|\}.$$

It is well-known that $P(x; F)$ is non-empty when $F \subset \mathbb{R}^n$ is closed. If we assume in addition that F is convex, than $P(x; F)$ is a singleton.

Proposition 5. [11, 15] Given any $a \in \mathbb{R}^n$ and $\gamma > 0$, Nesterov's smoothing approximation of $\varphi(x) = \|x - a\|_1$ has the representation

$$\varphi_\gamma(x) := \frac{1}{2\gamma}\left\|x - a\right\|^2 - \frac{\gamma}{2}\left[d_F\left(\frac{x-a}{\gamma}\right)\right]^2,$$

where F is the closed unit box of $\mathbb{R}^n$, that is, $F := \{x = (x^1, \ldots, x^n) \in \mathbb{R}^n \mid -1 \le x_i \le 1 \text{ for } i = 1, \ldots, n\}$. Moreover,

$$\begin{aligned}\nabla\varphi_\gamma(x) &= P\left(\frac{x-a}{\gamma}; F\right)\\ &= P\left(\frac{x-a}{\gamma}; F\right)\\ &= \max(-e, \min(\frac{x-a}{\gamma}, e)) \quad \text{component-wise,} \end{aligned} \tag{6}$$

where P is the Euclidean projection from $\left(\frac{x-a}{\gamma}\right)$ onto unit box F, and $e \in \mathbb{R}^n$ is a vector with one in each coordinate and zero elsewhere. In addition, $\varphi_\gamma(x) \le \varphi(x) \le \varphi_\gamma(x) + \frac{\gamma}{2}$.

3 Problems formulation

To define our problems, consider a set A of m data points, that is, $A = \left\{a^i \in R^n : i = 1, \ldots, m\right\}$ and k variable cluster centers denoted by $x^1, \ldots, x^k$. We model a two-level hierarchical clustering problem by choosing k separate cluster centers from which one is the headquarter that serves the centers. Other members of the data will be assigned to one of the cluster based on the ℓ_1 norm between the data points and centers. Thus, nodes are grouped into k variable centers by minimizing the ℓ_1 distances from all node to k centers. Then, a headquarter is a center that minimizes the overall distance of the network and also serves as a cluster center. Then, headquarter is defined to be mean of x^j for $j = 1, .., k$, that is, $\bar{x} = \frac{1}{k}\sum_{j=1}^k x^j$. This constraint limits the search region for headquarter to mean of selected centers or node near mean. Mathematically, the problem is defined as follows:

$$\begin{aligned} f(X) = \sum_{i=1}^m \min\left\{\|x^1 - a^i\|_1, ..., \|x^k - a^i\|_1, \|\bar{x} - a^i\|_1\right\}\\ + \sum_{j=1}^k \|x^j - \bar{x}\|_1 \end{aligned}$$

is minimized, where,

$$\bar{x} = \frac{1}{k}\sum_{j=1}^k x^j$$

In addition, to insure the centers are real node, the points $\bar{x}, x^1, x^2, ..., x^k$ should satisfy the following condition:

$$\min_{i=1,...,m} \|\bar{x} - a^i\|_1 + \sum_{j=1}^k \min_{i=1,...m} \|x^j - a^i\|_1 = 0$$

Thus, the problem is formulated as

$$\text{minimize}\left\{\sum_{i=1}^m \min_{j=1,...,k+1} \|x^j - a^i\|_1 + \sum_{j=1}^k \|x^j - \bar{x}\|_1\right\} \tag{7}$$

subject to

$$\sum_{j=1}^{k+1} \min_{i=1,...m} \|x^j - a^i\|_1 = 0, \tag{8}$$

where x^{k+1} in the summation is $\bar{x}$. The constraints in (8) are used to force the centers to lie on real node and to force headquarter to be on or near mean of the centers based on minimum distance.

We can write (7) as unconstrained problem using penalty parameter $\tau > 0$, as follows:

$$\begin{aligned}\text{minimize}\Bigg(\sum_{i=1}^m \min_{j=1,...,k+1} \|x^j - a^i\|_1 + \sum_{j=1}^k \|x^j - \bar{x}\|_1\\ + \tau\sum_{i=1}^{k+1} \min_{i=1,...,m} \|x^j - a^i\|_1\Bigg).\end{aligned}$$

Writing (9) as the sum and maximum of convex functions using the formula in (4) as follows:

$$\begin{aligned} f(X) = \textstyle\sum_{i=1}^m \sum_{j=1}^{k+1} \|x^j - a^i\|_1\\ - \textstyle\sum_{i=1}^m \max_{t=1,...,k+1} \sum_{j=1,j\neq t}^{k+1} \|x^j - a^i\|_1 + \sum_{j=1}^k \|x^j - \bar{x}\|_1\\ +\tau \textstyle\sum_{i=1}^m \sum_{j=1}^{k+1} \|x^j - a^i\|_1 - \tau \max_{t=1,...,k+1}\\ \textstyle\sum_{j=1,j\neq t}^{k+1} \|x^j - a^i\|_1. \end{aligned} \tag{9}$$

Expressing (9) as DC function, we have

$$\begin{aligned} f(X) &= (1+\tau)\sum_{i=1}^m \sum_{j=1}^{k+1} \|x^j - a^i\|_1 + \sum_{j=1}^k \|x^j - \bar{x}\|_1\\ &- \sum_{i=1}^m \max_{t=1,...,k+1} \sum_{j=1,j\neq t}^{k+1} \|x^j - a^i\|_1\\ &- \tau\sum_{j=1}^{k+1} \max_{r=1,...,m} \sum_{i=1,i\neq r}^{m} \|x^j - a^i\|_1, \end{aligned} \tag{10}$$

where

$$g(X) = (1+\tau)\sum_{i=1}^m \sum_{j=1}^{k+1} \|x^j - a^i\|_1 + \sum_{j=1}^k \|x^j - \bar{x}\|_1, \quad \text{and}$$

$$h(X) = \sum_{i=1}^{m} \max_{t=1,\dots,k+1} \sum_{j=1, j\neq t}^{k+1} \|x^j - a^i\|_1 + \tau \sum_{j=1}^{k+1} \max_{r=1,\dots,m} \sum_{i=1, i\neq r}^{m} \|x^j - a^i\|_1. \quad (11)$$

Since f is DC function based on Proposition 5 and ℓ_1 smoothing studied in [11], we obtain a Nesterov's approximation of $\|x-a\|_1$ as

$$\|x-a\|_1 := \frac{\gamma}{2}\left[\|\frac{x-a}{\gamma}\|^2 - [d_F(\frac{x-a}{\gamma})]^2\right].$$

The main goal is to minimize the partially smoothed objective given by,

$$\begin{aligned} f_\gamma(X) = {} & \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k+1}\|\frac{x^j-a^i}{\gamma}\|^2 + \sum_{j=1}^{k}\|\frac{x^j-\bar{x}}{\gamma}\|^2 \\ & -\frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k+1}[d_F(\frac{x^j-a^i}{\gamma})]^2 - \frac{\gamma}{2}\sum_{j=1}^{k}[d_F(\frac{x^j-\bar{x}}{\gamma})]^2 \\ & -\sum_{i=1}^{m}\max_{t=1,\dots,k+1}\sum_{j=1, j\neq t}^{k+1}\|x^j-a^i\|_1 \quad (12) \\ & -\tau\sum_{j=1}^{k+1}\max_{r=1,\dots,m}\sum_{i=1, i\neq r}^{m}\|x^j-a^i\|_1. \end{aligned}$$

That is minimize $\{f_\gamma(X) = g_\gamma(X) - h_\gamma(X)\}, X \in \mathbb{R}^{k\times n}$.
In addition, g_γ is the sum of convex functions defined as

$$g_\gamma(X) = g_{1\gamma}(X) + g_{2\gamma}(X) \quad (13)$$

where

$$g_{1\gamma}(X) = \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k+1}\|\frac{x^j-a^i}{\gamma}\|^2, \quad g_{2\gamma}(X) = \sum_{j=1}^{k}\|\frac{x^j-\bar{x}}{\gamma}\|^2.$$

And h_γ is also the sum of four convex functions defined as

$$h_\gamma(X) = h_{1\gamma}(X) + h_{2\gamma}(X) + h_{3\gamma}(X) + h_{4\gamma}(X), \quad (14)$$

where

$$\begin{aligned} h_{1\gamma}(X) &= \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k+1}[d_F(\frac{x^j-a^i}{\gamma})]^2 \\ h_{2\gamma}(X) &= \frac{\gamma}{2}\sum_{j=1}^{k}[d_F(\frac{x^j-\bar{x}}{\gamma})]^2, \\ h_{3\gamma}(X) &= \sum_{i=1}^{m}\max_{t=1,\dots,k+1}\sum_{j=1, j\neq t}^{k+1}\|x^j-a^i\|_1, \\ h_{4\gamma}(X) &= \tau\sum_{j=1}^{k+1}\max_{r=1,\dots,m}\sum_{i=1, i\neq r}^{m}\|x^j-a^i\|_1. \end{aligned}$$

For the calculation of gradient and sub-gradient, consider a data matrix A with a^i, $i = 1, \dots, m$, in the i^{th} row and a variable matrix X with $x^j, j = 1, 2, \dots, k+1$ in the j^{th} row.

Since X and A belongs to a linear space of real matrices, we can apply inner product such that

$$\langle X, A\rangle = \text{trace}(X^T A) = \sum_{i=1}^{n}\sum_{j=1}^{k} x_{ij}a_{ij}.$$

And the Frobenius norm on $\mathbb{R}^{k\times m}$ is given by

$$\|A\|_F = \sqrt{\langle A, A\rangle} = \sqrt{\sum_{j=1}^{k}\langle a^j, a^j\rangle} = \sqrt{\sum_{j=1}^{k}\|a^j\|^2}. \quad (15)$$

To calculate the gradient of g_γ in (13), let X be of size $(k+1)\times n$ is variable matrix. Then,

$$\begin{aligned} g_{1\gamma}(X) &:= \frac{(1+\tau)}{2\gamma}\sum_{i=1}^{m}\sum_{j=1}^{k+1}\|x^j-a^i\|^2, \\ &= \frac{(1+\tau)}{2\gamma}\sum_{i=1}^{m}\sum_{j=1}^{k}\left[\|x^j\|^2 - 2\langle x^j, a^i\rangle + \|a^i\|^2\right], \\ &= \frac{(1+\tau)}{2\gamma}\left[m\|X\|_F^2 - 2\langle X, E_{km}A\rangle + k\|A\|_F^2\right], \end{aligned}$$

where $E_{km} \in \mathbb{R}^{k+1\times m}$ is a matrix of all ones. As $g_{1\gamma}$ is smooth, then

$$\nabla_x g_{1\gamma}(X) = \frac{(1+\tau)}{\gamma}\left[mX - B\right], \quad \text{where} \quad B = E_{km}A.$$

Again consider $g_{2\gamma}$ which is differentiable function,

$$\begin{aligned} g_{2\gamma}(X) &:= \frac{1}{2\gamma}\sum_{j=1}^{k}\|x^j-\bar{x}\|^2, \\ &= \frac{1}{2\gamma}\sum_{j=1}^{k}\left[\|x^j\|^2 - 2\langle x^j, \bar{x}\rangle + \langle x^j, \bar{x}\rangle\right], \\ &= \frac{1}{2\gamma}\left[\|X\|_F^2 - \frac{2}{k}\langle X, E_{kk}X\rangle + \frac{1}{k}\langle X, E_{kk}X\rangle\right], \end{aligned}$$

where E_{kk} is a $k\times k$ matrix with elements all ones. Then, the gradients of $g_{2\gamma}$ are given by

$$\begin{aligned} \nabla_x g_{2\gamma}(X) &= \frac{1}{\gamma}\left[X - \frac{1}{k}E_{kk}X\right], \\ &= \frac{1}{\gamma}\left[X - HX\right], \text{where } H = \frac{1}{k}E_{kk}. \end{aligned}$$

Next, we focus on $X \in \partial g^*(Y)$ where g^* is a Fenchel conjugate defined in (2) and can be calculated using the fact that $X \in$

$\partial g^*(Y) \Leftrightarrow Y \in \partial g(X)$. Since g_γ is differentiable. Thus,

$$\begin{aligned}
\nabla_x g_\gamma(X) &= \nabla_x g_{1\gamma}(X) + \nabla_x g_{2\gamma}(X),\\
Y &= \frac{(1+\tau)}{\gamma}\left[mX - B\right] + \frac{1}{\gamma}\left[X - HX\right],\\
&= \left[\frac{(1+\tau)}{\gamma}m + \frac{1}{\gamma}\left[\mathbb{I} - H\right]\right]X - \left[\frac{(1+\tau)}{\gamma}B\right],\\
&= \frac{1}{\gamma}\left[(1+\tau)m + \mathbb{I} - H\right]X - \left[\frac{(1+\tau)}{\gamma}B\right],\\
&= \frac{1}{\gamma}\left[(1+\tau)m + 1]\mathbb{I} - H\right]X - \left[\frac{(1+\tau)}{\gamma}B\right],\\
&= \frac{1}{\gamma}\left[a\mathbb{I} - bH\right]X - \left[\frac{(1+\tau)}{\gamma}B\right],
\end{aligned}$$

where $a = (1+\tau)m + 1$ and $b = 1$.

Let $N = a\mathbb{I} - bH$, then N is invertible as $N^{-1} = \alpha\mathbb{I} + \beta H$ where

$$\alpha = \frac{1}{a} = \frac{1}{(1+\tau)m+1} \quad \text{and}$$

$$\beta = \frac{b}{a[a+bk]} = \frac{1}{(1+\tau)m+1[(1+\tau)m+1+k]},$$

(see Lemma 5.1 of [8]). Therefore,

$$X = [\alpha\mathbb{I} - \beta H]\left[\gamma Y_x + (1+\tau)B\right]. \tag{16}$$

Next, we find the sub-gradient in (14) and this can be done by search $Y \in \partial h_\gamma(X)$. Given a smooth functions $h_{1\gamma}$ and $h_{2\gamma}$, the partial gradient at x^j for $j = 1, \ldots, k+1$ is

$$h_{1\gamma} = \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k+1}\left[d_F(\frac{x^j - a^i}{\gamma})\right]^2$$

$$\frac{\partial h_{1\gamma}}{\partial x^j}(X) = (1+\tau)\gamma\sum_{i=1}^{m}\left[\frac{x^j - a^i}{\gamma} - P(\frac{x^j - a^i}{\gamma}; F)\right]. \tag{17}$$

Thus, $\nabla h_{1\gamma}(X))$ is a matrix with dimension $(k+1)\times n$ with j^{th} row is $\frac{\partial h_{1\gamma}}{\partial x^j}(X)$.

The gradients of $h_{2\gamma} = \frac{\gamma}{2}\sum_{j=1}^{k}\left[d_F(\frac{x^j - \bar{x}}{\gamma})\right]^2$ at X are given by

$$\begin{aligned}
\frac{\partial h_{2\gamma}}{\partial x^j}(X) &= \frac{x^j - \bar{x}}{\gamma} - P(\frac{x^j - \bar{x}}{\gamma}; F)\\
&\quad - \frac{1}{k}\sum_{\ell=1}^{k}\left[\frac{x^\ell - \bar{x}}{\gamma} - P(\frac{x^\ell - \bar{x}}{\gamma}; F)\right].
\end{aligned} \tag{18}$$

The projections in (17) and (18) are the Euclidean projection from $v \in \mathbb{R}^n$ onto a unit closed box F which are defined as

$$P(v, F) = \max(-e, \min(v, e)).$$

where $e \in \mathbb{R}^n$ is a vector with one in each coordinate and zero elsewhere.

Since we use ℓ_1 norm, next we illustrate how to find the sub-gradient $Y \in \partial h_\gamma(X)$ for the case where F is the closed unit box in $\mathbb{R}^n$.

For a given $x \in \mathbb{R}$, we define

$$\text{sign}(x) := \begin{cases} 1 & \text{if } x > 0,\\ 0 & \text{if } x = 0,\\ -1 & \text{if } x < 0. \end{cases}$$

Then, we define $\text{sign}(x) := (\text{sign}(x_1), \ldots, \text{sign}(x_n))$ for $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Note that the sub-gradients of $f(x) = \|x\|_1$ at $x \in \mathbb{R}^n$ are $s_i = \text{sign}(x)$ if $x_i \neq 0$ and $s_i \in [-1, 1]$ if $x_i = 0$.

The sub-gradients of the non-smooth functions $h_{3\gamma}$ and $h_{4\gamma}$ are calculated as the sub-differential of point-wise maximum functions,

$$h_{3\gamma} := \sum_{i=1}^{m}\max_{r=1,\ldots,k}\sum_{j=1, j\neq r}^{k}\|x^j - a^i\|_1 = \sum_{i=1}^{m}\phi_i(\mathbf{X}),$$

where, for $i = 1, \ldots, m$,

$$\phi_i(\mathbf{X}) := \max\left\{\phi_{ir}(\mathbf{X}) = \sum_{j=1, j\neq r}^{k}\|x^j - a^i\|_1, \quad r = 1, \ldots, k\right\}.$$

To do this, for each $i = 1, \ldots, m$, we first find $\mathbf{U}_i \in \partial\phi_i(\mathbf{X})$ according to Lemma 3. Then, we define $\mathbf{U} := \sum_{i=1}^{m}\mathbf{U}_i$ to get a sub-gradient of the function $h_{3\gamma}$ at $\mathbf{X}$ by the sub-differential sum rule. To accomplish this goal, we first choose an index r^* from the index set $\{1, \ldots, k\}$ such that

$$\phi_i(\mathbf{X}) = \phi_{ir^*}(\mathbf{X}) = \sum_{j=1, j\neq r^*}^{k}\|x^j - a^i\|_1.$$

Using the familiar sub-differential formula of the ℓ_1 norm function, the j^{th} row u_i^j for $j \neq r^*$ of the matrix $\mathbf{U}_i$ is determined as follows:

$$u_i^j := \text{sign}(x^j - a^i) = \begin{cases} 1 & \text{if } x^j > a^i,\\ 0 & \text{if } x^j = a^i,\\ -1 & \text{if } x^j < a^i. \end{cases}$$

The r^{*th} row of the matrix $\mathbf{U}_i$ is $u_i^{r^*} := 0$.

Similarly, the sub-gradient of $h_{4\gamma}$ is given by

$$h_{4\gamma} := \tau\sum_{j=1}^{k}\max_{s=1,\ldots,m}\sum_{i=1, i\neq s}^{m}\|x^j - a^i\|_1 = \tau\sum_{j=1}^{k}\psi_j(\mathbf{X}),$$

where, for $j = 1, \ldots, k$,

$$\psi_j(\mathbf{X}) := \max\left\{\psi_{js}(\mathbf{X}) = \sum_{i=1, i\neq s}^{m}\|x^j - a^i\|_1, \quad s = 1, \ldots, m\right\}.$$

To do this, for each $j = 1, \ldots, k$, we first find $\mathbf{W}_j \in \partial\psi_j(\mathbf{X})$. Then, we define $\mathbf{W} := \tau\sum_{j=1}^{k}\mathbf{W}_j$ to get a sub-gradient of the function $h_{4\gamma}$ at $\mathbf{X}$ by the sub-differential sum rule. To accomplish this goal, we first choose an index s^* from the index set $\{1, \ldots, m\}$ such that

$$\psi_j(\mathbf{X}) = \psi_{js^*}(\mathbf{X}) = \sum_{j=1, j\neq s^*}^{k}\|x^j - a^i\|_1.$$

The s^{*th} row of the matrix $\mathbf{W}_j$ is $w^j_{s*} := 0$.
Thus, the sub-gradient of h_{4_γ} is defined as

$$\frac{\partial h_{4_\gamma}}{\partial x^j} := \tau \mathbf{W}.$$

From the sub-gradient calculated above we have,

$$Y = \frac{\partial h_{1\gamma}}{\partial x^j}(X) + \frac{\partial h_{2\gamma}}{\partial x^j}(X) + \frac{\partial h_{3\gamma}}{\partial x^j}(X) + \frac{\partial h_{4\gamma}}{\partial x^j}(X) \quad (19)$$

Now, we have in position to implement DCA algorithm that will solve the problem as shown in DCA Algorithm 2.

Input : $A, X_0, \tau_0, \gamma_0, N \in \mathbb{N}$,
while *stopping criteria for* $(\gamma, \tau, \varepsilon)$ *not reached* **do**
  $\alpha \leftarrow \frac{1}{(1+\tau_1)m+1}$,
  $\beta \leftarrow \frac{1}{(1+\tau_1)m+1[(1+\tau_1)m+1+k]}$,
  for $k = 1, \ldots, N$ **do**
    Find $Y_k \in \partial h(x_{k-1})$, (19)
    $X_k \leftarrow [\alpha\mathbb{I} + \beta H]\left[\gamma Y_k + (1+\tau)B\right]$, (16)
  end
  update γ and τ,
end
Output: X_N.

Algorithm 2. Bi-level hierarchical clustering.

4 Simulation results

The numerical simulation was performed on an HP laptop with an Intel(R) Core(TM) i7-8565U at 1.80 GHz 1.99 GHz processor, 8.00 GB RAM with MATLAB version R2017b. Various parameters were used during the simulation, among others we used a large increasing penalty parameter τ and a decay smoothing parameter γ. These parameters are updated during iteration as in [6]; $\tau_{i+1} = \sigma_1\tau_i, \sigma_1 > 1$ and $\gamma_{i+1} = \sigma_2\gamma_i, 0 < \sigma_2 < 1$ and $\epsilon = 1e{-}6$. We chose the initial penalty parameter ($\tau_0 = e^{-6}$) and the initial smoothing parameter $\gamma_0 = 1$. In addition, after varying the parameters, we chose $\sigma_1 \leq 16e^9$ as the growth factor of the penalty parameter, $\sigma_2 = 0.5$ the decrease factor of the smoothing parameter, and the stopping criterion $\frac{\|X_{k+1}-X_k\|_F}{\|X_k\|_F+1} \leq \epsilon$ for inner for loop. To implement the algorithms, we used randomly selected default cluster centers from the datasets.

The performance of the DCA Algorithm 2 was tested with different datasets. We first tested the algorithm on a small dataset taken from [8], and the result shows that it converges to the same cluster centers as in [8] with a different objective value due to the ℓ_1 norm. Since the ℓ_1 distance is greater than or equal to the Euclidean distance, it depends on the data points. As shown in Table 1, the algorithm converges to the optimal point approximately 85.71%. This means that out of 7 iterations, 6 of them converge to the same objective valve.

Second, we tested the proposed algorithm with EIL76 (The 76 City Problem) datasets taken from [7] with four clusters, one of which serves as HQ, which converge to near-optimal cluster centers in a reasonable time compared to study [8, 13] (see Figure 1).

It is also observed in the EIL76 (The 76 City Problem) data which converge to the same or close cluster centers with higher objective cost, fewer iterations, and almost the same time compared to the study of [8] iterated using MATLAB (see Table 2).

Third, we applied our proposed algorithm to a GPS data from 142 cities and towns in Ethiopia with more than 7,000 inhabitants, including 65 in Oromia regional state. We tested the algorithm with 65 nodes, 4 cluster centers, one of which serves as HQ, and 142 nodes with six clusters (see Figures 2, 3), which converge 86% to the optimal solution. This means that out of 7 iterations, 6 of them converge to the near-optimal values shown in Tables 3, 4.

Fourth, we tested the proposed algorithm with PR1002 (The 1002 City Problem) datasets taken from [7] with seven clusters, one of which serves as HQ, which converge to near-optimal cluster centers in a reasonable time compared to study [8, 13] (see Table 5 and Figure 4).

TABLE 1 Ten iterations for the 15 point test dataset.

Dataset	Cost	Time	Iteration	Centers	Data size
Test data	19.142400	1.101548	80	3	15
Test data	19.189316	1.980110	80	3	15
Test data	19.232334	1.839473	80	3	15
Test data	19.232334	1.839473	80	3	15
Test data	19.213091	1.807942	80	3	15
Test data	19.210684	1.816595	80	3	15
Test data	20.904108	2.005234	80	3	15
Test data	19.199307	1.617948	80	3	15
Test data	19.191683	1.880256	80	3	15
Test data	19.190641	1.795777	80	3	15

For $\gamma_0 = 1, \tau_0 = 10^{-6}, \sigma_1 = 16e + 9$, and $\sigma_2 = 0.75$.

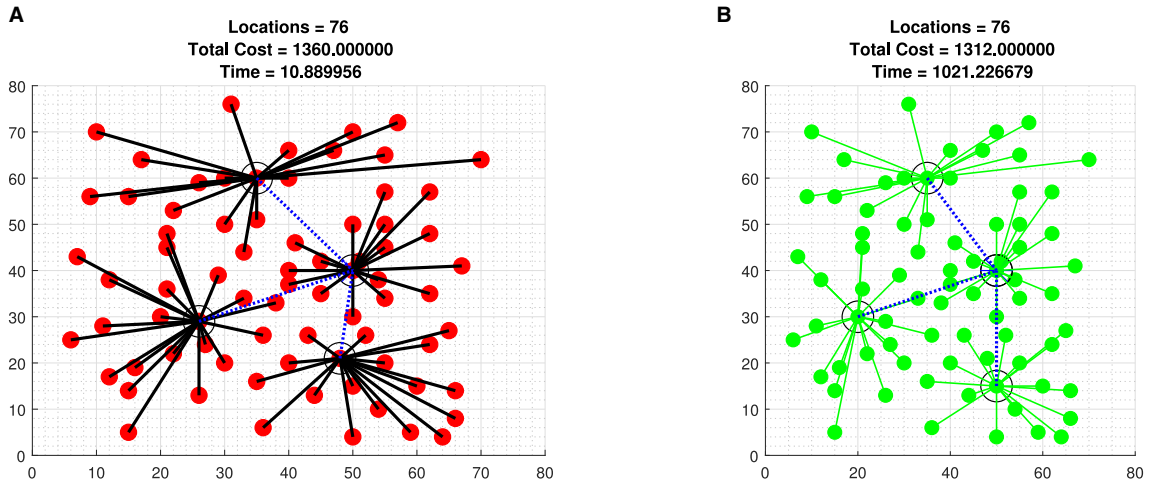

FIGURE 1
Optimal cost of EIL76 (The 76 City Problem) data taken from [7] with four clusters one serves as HQ. **(A)** A EIL76 data with Algorithm 2. **(B)** A EIL76 data with brute-force iteration.

TABLE 2 Ten iterations for EIL76 dataset.

Dataset	Cost	Time	Iteration	Centers	Data size
EIL76	1,370.002095	4.966008	300	4	76
EIL76	1,360.029069	6.577355	300	4	76
EIL76	1,360.034103	5.752709	300	4	76
EIL76	1,360.034103	5.789645	300	4	76
EIL76	1,360.034103	6.184165	300	4	76
EIL76	1,362.015115	11.135013	300	4	76
EIL76	1,360.040060	6.680976	300	4	76
EIL76	1,360.034103	7.120168	300	4	76
EIL76	1,360.034103	5.275355	300	4	76
EIL76	1,360.034103	6.191321	300	4	76

For $\gamma_0 = 1$, $\tau_0 = 10^{-6}$, $\sigma_1 = 16e + 9$, and $\sigma_2 = 0.5$.

To show how the objective functions improved with iteration, we include a plot of the first few iterations of Figures 3, 4, which shows the dynamics of the algorithm (see Figures 3A, B, 4A, B).

In general, since the algorithm is a modified DCA and DCA is a local search algorithm, there is no guarantee that our algorithms converge to the global optimal solution. However, we compared our result with ℓ_2 norm in [8, 13], and it shows that our proposed algorithm converges with fewer iteration but relatively the same computational time for data iterated with MATLAB in [8]. In addition, we compared our result with brute-force generated solutions for datasets with fewer nodes (see Figures 1A, B, 2A, B) which converge to a near-optimal value with reasonable time compared to the brute-force iterations.

We expect that our method used in this study to solve the two-level clustering problem with the ℓ_1 norm is less sensitive to outliers compared to the ℓ_2 norm, which minimizes possible clustering errors. In addition, it can be used to solve other non-smooth and non-convex optimization problems in signal processing, such as image pixel clustering for image segmentation and compressed sensing.

For the following tables, we conducted an experiment with fixed iteration numbers for each dataset and initial cluster centers were randomly selected from the datasets. The cost is obtained by minimizing Equation (7).

Figure 5 shows the optimal cost of test data taken from [8] with optimal clusters centers and HQ,

$$X = \begin{pmatrix} 7.0000 & 2.0000 \\ 4.5000 & 2.0000 \\ 2.0000 & 2.0000 \end{pmatrix}, \quad HQ = (4.5000 \quad 2.0000)$$

In Figure 1 the selected cluster centers and HQ are

$$X = \begin{pmatrix} 26.0000 & 29.0000 \\ 35.0000 & 60.0000 \\ 50.0000 & 40.0000 \\ 48.0000 & 21.0000 \end{pmatrix} \quad HQ = (50.0000 \;\; 40.0000),$$

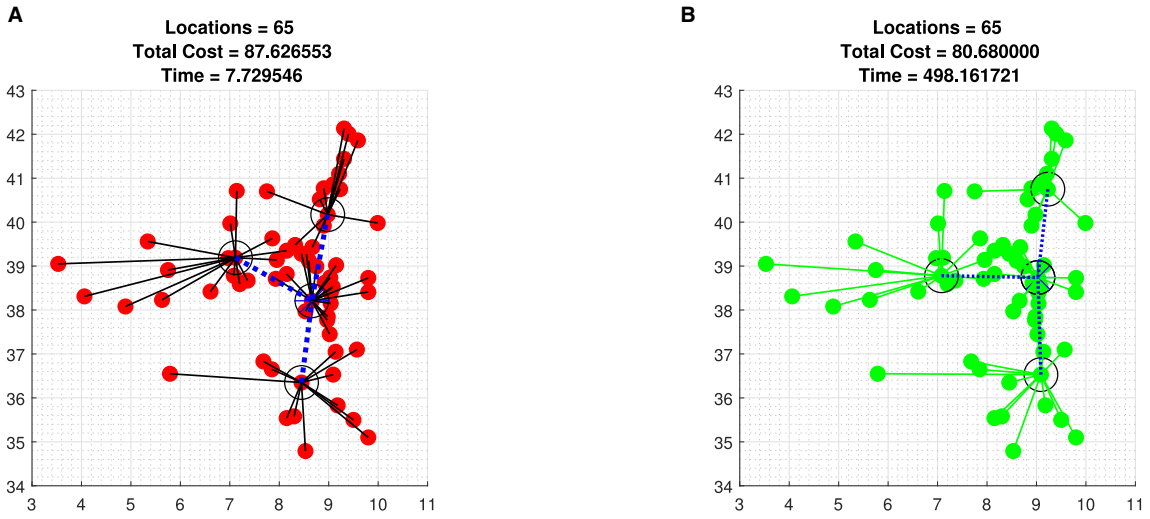

FIGURE 2
Datasets of 65 Oromia regional cities and towns with four clusters one serves as HQ. **(A)** Sixty-five data points using Algorithm 2. **(B)** Sixty-five data points with brute-force iteration.

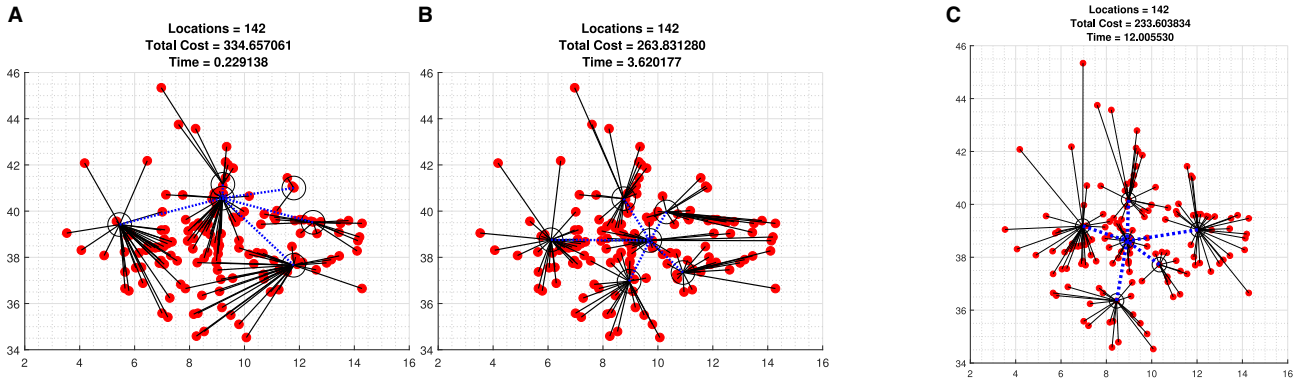

FIGURE 3
Datasets of 142 Ethiopian towns and cities with six clusters one serves as HQ. **(A)** Beginning of iterations. **(B)** Few iterations. **(C)** Optimal iterations.

TABLE 3 **Ten iterations for the 65 point test dataset.**

Dataset	Cost	Time	Iteration	Centers	Data size
A 65 points data	87.689552	7.639907	250	4	65
A 65 points data	87.889552	7.655031	250	4	65
A 65 points data	89.889552	7.542318	250	4	65
A 65 points data	87.889552	7.520294	250	4	65
A 65 points data	87.389552	8.101386	250	4	65
A 65 points data	87.689552	7.324263	250	4	65
A 65 points data	87.789552	7.289665	250	4	65
A 65 points data	87.789552	7.545756	250	4	65
A 65 points data	87.389552	7.399864	250	4	65
A 65 points data	87.089552	7.460620	250	4	65

For $\gamma_0 = 1$, $\tau_0 = 10^{-6}$, $\sigma_1 = 16e + 9$, and $\sigma_2 = 0.75$.

TABLE 4 Ten iterations for the 142 point test dataset.

Dataset	Cost	Time	Iteration	Centers	Data size
A 142 points data	233.793443	13.211982	300	6	142
A 142 points data	233.793443	13.967717	300	6	142
A 142 points data	233.793443	13.197907	300	6	142
A 142 points data	233.793443	13.857134	300	6	142
A 142 points data	233.893443	15.259091	300	6	142
A 142 points data	233.893443	12.552241	300	6	142
A 142 points data	241.793443	13.426098	300	6	142
A 142 points data	233.783342	12.750388	300	6	142
A 142 points data	233.783241	10.986166	300	6	142
A 142 points data	233.783242	12.123523	300	6	142

For $\gamma_0 = 1$, $\tau_0 = 10^{-6}$, $\sigma_1 = 16e + 9$, and $\sigma_2 = 0.5$.

TABLE 5 Ten iterations for PR1002 dataset.

Dataset	Cost	Time	Iteration	Centers	Data size
PR1002	2.037468e+6	55.747814	350	7	1002
PR1002	2.036560e+6	49.831589	350	7	1002
PR1002	2.036560e+6	47.624006	350	7	1002
PR1002	2.034309e+6	56.673023	350	7	1002
PR1002	2.034309e+6	62.799942	350	7	1002
PR1002	2.034309e+6	58.704935	350	7	1002
PR1002	2.034309e+6	61.162610	350	7	1002
PR1002	2.033309e+6	65.173468	350	7	1002
PR1002	2.033309e+6	65.183165	350	7	1002
PR1002	2.035309e+6	56.747571	350	7	1002

For $\gamma_0 = 1600$, $\tau_0 = 10^{-6}$, $\sigma_1 = 8000$, and $\sigma_2 = 0.5$.

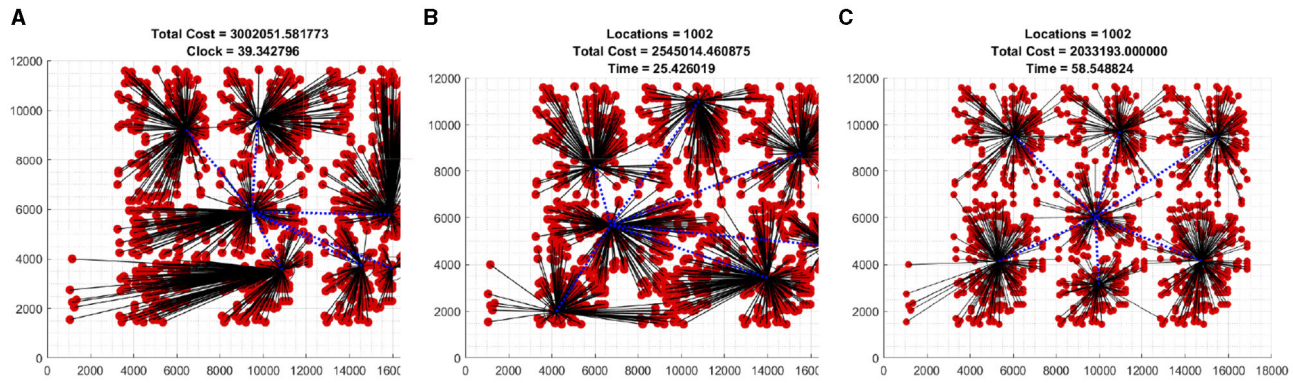

FIGURE 4
Datasets of PR1002 (The 1002 City Problem) taken from [7] with seven clusters and one serving as HQ. **(A)** Beginning of iteration. **(B)** Few iteration. **(C)** Optimal iteration.

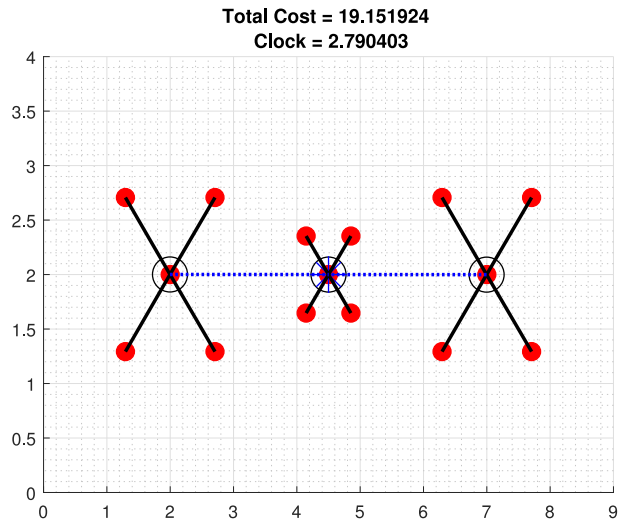

FIGURE 5
Fifteen point test dataset with three clusters and one serves as HQ.

In Figure 2 the selected cluster centers and HQ are

$$X = \begin{pmatrix} 8.4500 & 36.3500 \\ 8.8990 & 39.9171 \\ 8.6607 & 38.2124 \\ 6.6100 & 38.4200 \end{pmatrix} \text{ and } HQ = (8.6607 \;\; 38.2124).$$

In Figure 3 the selected cluster centers and HQ are

$$X = \begin{pmatrix} 10.3400 & 37.7199 \\ 8.4500 & 36.3500 \\ 6.9595 & 39.1795 \\ 8.9131 & 38.6186 \\ 8.9808 & 40.1709 \\ 12.0400 & 39.0400 \end{pmatrix} \text{ and } HQ = (8.9156 \;\; 38.6189).$$

For this particular dataset, we used $\gamma_0 = 1600$ and $\sigma_1 = 8000$. In Figure 4 the selected cluster centers and HQ are

$$X = \begin{pmatrix} 5218.0000 & 4090.0000 \\ 5923.0000 & 9557.0000 \\ 1083.9000 & 9857.0000 \\ 1473.5000 & 4145.0000 \\ 9977.0000 & 3008.0000 \\ 1547.1000 & 9522.0000 \\ 9892.0000 & 6023.0000 \end{pmatrix} \text{ and } HQ = (9891.6000 \;\; 6023.0000).$$

5 Conclusion

In this study, we used a continuous formulation of discrete two-level hierarchical clustering, where the distance between two data points is measured by the ℓ_1 norm. As a result, it became non-smooth and non-convex, on which Nesterov's smoothing and DC-based algorithms were implemented. We observe that parameter selection is the decisive factor in terms of accuracy and speed of convergence of our proposed algorithms. The performance of Algorithm 2 highly depends on the initial values set to the penalty and smoothing parameter.

The algorithm was tested with real and known source datasets of different sizes in MATLAB. Starting from different random initial cluster centers, the algorithm converges to a near-optimal value in a reasonable time. As a result, improved iteration time for large-scale problems and convergence to a near-optimal solution were observed.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

AG: Investigation, Software, Writing – original draft, Conceptualization, Data curation, Formal analysis, Funding acquisition, Methodology, Project administration, Resources, Supervision, Validation, Visualization, Writing – review & editing. LO: Conceptualization, Formal analysis, Supervision, Validation, Writing – review & editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Acknowledgments

Authors are grateful to the referees and handling editor for their constructive comments.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fams.2024.1445390/full#supplementary-material

References

1. An LTH, Belghiti MT, Tao PD. A new efficient algorithm based on DC programming and DCA for clustering. *J Global Optimizat*. (2007) 37:593–608. doi: 10.1007/s10898-006-9066-4

2. Bagirov AM, Taheri S, Ugon J. Nonsmooth DC programming approach to the minimum sum-of-squares clustering problems. *Pattern Recognit*. (2016) 53:12–24. doi: 10.1016/j.patcog.2015.11.011

3. Bagirov A. Derivative-free methods for unconstrained nonsmooth optimization and its numerical analysis. *Investigacao Operacional*. (1999) 19:75–93.

4. Bajaj A, Mordukhovich BS, Nam NM, Tran T. Solving a continuous multifacility location problem by DC algorithms. *Optimizat Meth Softw*. (2022) 37:338–60. doi: 10.1080/10556788.2020.1771335

5. Barbosa GV, Villas-Boas SB, Xavier AE. Solving the two-level clustering problem by hyperbolic smoothing approach, and design of multicast networks. In: *The 13th World Conference on Transportation Research was organized on July 15–18, 2013 by COPPE - Federal University of Rio de Janeiro, Brazil*. WCTR RIO (2013).

6. Fita A, Geremew W, Lemecha L. A DC optimization approach for constrained clustering with l1-norm. *Palest J Mathem*. (2022) 11:3.

7. Reinelt G. TSPLIB: A traveling salesman problem library. *ORSA J Comp*. (1991) 3:376–84. doi: 10.1287/ijoc.3.4.376

8. Geremew W, Nam NM, Semenov A, Bogniski V, Psailio E. A DC programming approach for solving multicast network design problems via the Nesterov smoothing technique. *J Global Optimizat*. (2018) 72:705–29. doi: 10.1007/s10898-018-0671-9

9. Jia L, Bagirov A, Ouveysi I, Rubinov M. Optimization based clustering algorithms in Multicast group hierarchies. In: *Proceedings of the Australian Telecommunications, Networks and Applications Conference (ATNAC)* (2003).

10. Le Thi HA, Pham Dinh T. DC programming and DCA: thirty years of developments. *Mathemat Program*. (2018) 169:5–68. doi: 10.1007/s10107-018-1235-y

11. Mau Nam N, Hoai An LT, Giles D, An NT. Smoothing techniques and difference of convex functions algorithms for image reconstructions. *Optimization*. (2020) 69:1601–33. doi: 10.1080/02331934.2019.1648467

12. Mordukhovich BS, Nam NM. *An Easy Path to Convex Analysis and Applications*. Cham: Springer. (2014).

13. Nam NM, Geremew W, Reynolds S, Tran T. Nesterov's smoothing technique and minimizing differences of convex functions for hierarchical clustering. *Optimizat Lett*. (2018) 12:455–73. doi: 10.1007/s11590-017-1183-0

14. Nam NM, Rector RB, Giles D. Minimizing differences of convex functions with applications to facility location and clustering. *J Optim Theory Appl*. (2017) 173:255–78. doi: 10.1007/s10957-017-1075-6

15. Nesterov Y. Smooth minimization of non-smooth functions. *Mathem Program*. (2005) 103:127–52. doi: 10.1007/s10107-004-0552-5

16. Nesterov Y. *Lectures on Convex Optimization*. Cham: Springer. (2018). doi: 10.1007/978-3-319-91578-4

17. Nguyen PA, Le Thi HA. DCA approaches for simultaneous wireless information power transfer in MISO secrecy channel. *Optimizat Eng*. (2023) 24:5–29. doi: 10.1007/s11081-020-09570-3

18. Rockafellar R. *Convex Analysis*. Princeton: Princeton University Press (1970).

19. Tao PD, An LTH. A DC optimization algorithm for solving the trust-region subproblem. *SIAM J Optimizat*. (1998) 8:476–505. doi: 10.1137/S1052623494274313

20. Tao PD, An LH. Convex analysis approach to DC programming: theory, algorithms and applications. *Acta Mathematica Vietnamica*. (1997) 22:289–355.

21. An LTH, Minh LH, Tao PD. Optimization based DC programming and DCA for hierarchical clustering. *Eur J Operation Res*. (2007) 183:1067–85. doi: 10.1016/j.ejor.2005.07.028

22. An LTH, Minh LH, Tao PD. New and efficient DCA based algorithms for minimum sum-of-squares clustering. *Pattern Recogn*. (2014) 47:388–401. doi: 10.1016/j.patcog.2013.07.012