Palestine Journal of Mathematics

Vol. 11(3)(2022) , 337–347

A DC Optimization Approach for Constrained Clustering with ℓ_1-Norm.

Adugna Fita, Wondi Geremew and Legesse Lemecha

Communicated by Ekeland Ivar

MSC 2010 Classifications: Primary 49J52, Secondary 49J53, 90C31.

Keywords and phrases: Clustering, DC programming, DC Algorithm, ℓ_1-norm, Nesterov's smoothing.

The authors like to thanks Adama Science and Technology University (ASTU) who supported this work with grant number ASTU/SP-R/002/19.

Abstract One of the challenges in optimizing clustering problems is requirement of differentiability. Clustering is a popular approach that classifies a given data into different groups , based on some common properties. It is a basic foundation of machine learning , facility location and image processing. Although the problem is nonsmooth, we used Nesterov partial smoothing technique to approximate nondifferentiable convex functions by smooth convex functions with Lipschitz continuous gradients. In this paper, we mainly focused in modelling and solving clustering problems that identify some nodes as cluster centers among others and minimize the overall ℓ_1 distance of the clusters. In addition, since the algorithm starts with any initial cluster centers penalty parameter is used to push centers to real node. As a result, a DCA based algorithms were implemented that find optimal cluster centers in reasonable iteration time.

1 Introduction

Currently, clustering is an area that mostly studied and applied to many branches of mathematics and computer science. The majority of clustering problems are nonsmooth and nonconvex that is not ruled by gradient decent algorithms. Besides, most clustering problems are discrete and combinatorial in nature which is challenging to obtain optimality.
The combination of Nesterov's smoothing technique [18] , DC programming and DCA [16] introduced an efficient platform to study nonconvex and nonsmooth problem in optimization. DCA has applied to facility location, compressed sensing, and imaging, supply chain management and telecommunication successfully [15, 16, 17, 19]. In this regard a number of works were done on clustering, among them, the minimum sum of squares clustering problem [6], the bilevel hierarchical clustering problem [14] and the multicast network design problem [10]. More recently, solving multifacility location problems by DC algorithms were studied in [17]. A similar problem was also investigated in [3] using different approach. The significant difference between problem studied in [17] and [3] is a squared Euclidean norm is used in the former while Euclidean norm is applied in the later case. An algorithms that were employed to solve those problems are mostly meta-heuristic and difficult to analysis optimality. In [3] DCA which was developed in [4, 13] was utilized by replacing ℓ_2-norm by squared ℓ_2-norm and applied to higher dimensional problems. DC algorithm has laid an efficient foundation in solving different nonconvex problems in clustering, see for example in [1, 2, 7, 9] and references cited therein. Recently, the authors in [10, 14] implemented a new way on Nesterov's and DC Algorithm to overcome the problem in [5].The idea of using Nesterov's smoothing techniques overcomes the drawback of applying DCA stated in [3] as the model is not appropriate to implement the DCA. In most applied problems, ℓ_1 distance measures can give a better approximation of the reality than Euclidean distance. In this paper, we further study clustering problem by modifying the objective and constraints functions using ℓ_1 norm. Since ℓ_1 norm is nonsmooth, we employed Nesterov's partial smoothing techniques and appropriate DC decomposition that helps to apply DC Algorithm (DCA). Besides, the change made to the objective function and constraints, the centers are forced to lie on real nodes in the datasets that minimize the overall distance. Thus, we used penalty parameters to convert constrained problem to unconstrained one.

The paper is organized as follows. In Section 2, we present basic tools of convex analysis that is applied to DC functions and DCA. In Sections 3, mainly focus on clustering problem formulation and analysis to develop DCA algorithms that solve the problem based on Nesterov's smoothing technique. In Section 4, numerical simulation results with some artificial data are demonstrated and some concluding remarks are presented in Section 5.

2 Basics of Convex functions

This section presents basic concepts of convex functions and convex analysis that will be used to study DC functions and convergence of DCA which is used in subsequent sections.

Definition 2.1. [8] A vector $v \in \Omega \subset \mathbb{R}^n$ is a subgradient of a convex function $f : \Omega \to \bar{\mathbb{R}}$ 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 subgradients is known as the subdifferential of f at $\bar{x}$ and is denoted by $\partial f(\bar{x}) = \{v \in \mathbb{R}^n \mid v \text{ satisfies definition 2.1}\}$.

We present the following consequentive results without technical proofs. Detailed proofs are given in [8].

Theorem 2.2. *[8] Let $f_j : \mathbb{R}^n \to \bar{\mathbb{R}}$ be proper extended real-valued convex functions and the intersection of relative interior of domain $\bigcap_{j=1}^n ri(dom(f_j)) \neq \emptyset$ where $n \geq 2$. Then for all $\bar{x} \in \bigcap_{j=1}^n dom(f_j)$*

$$\partial \left(\sum_{j=1}^{n}) f_j(\bar{x}) \right) = \sum_{j=1}^{n} \partial f_j(\bar{x})$$

The Maximum Function is defined as the pointwise maximum of convex functions. For $j = 1, 2, 3, ..., n$, let functions $f_j : \mathbb{R}^n \to \mathbb{R}$ be closed and convex. Then the maximum function

$$f(x) := \max_{j=1,...,n} f_j(x) = \max \{f_1(x), f_2(x), ..., f_n(x)\}$$

is also closed and convex.
The Minimum Function $f(x)$, defined by

$$f(x) := \min_{j=1,...n} f_j(x) = \min \{f_1(x), f_2(x), ..., f_n(x)\}$$

may not be convex.
Subdifferential of most class of nondifferentiable convex functions is defined as the pointwise maximum of convex functions.

Lemma 2.3. *Let $f_1, f_2, ..., f_n : \mathbb{R}^n \to \bar{\mathbb{R}}$ be proper extended real-valued convex functions and let $f(x) = \max \{f_1(x), f_2(x), ..., f_n(x)\}$. if $x \in \bigcap_{j=1}^n int(dom(f_j))$, then*

$$\partial f(x) = co \left(\bigcup_{j \in I(x)} \partial f_k(x) \right) \quad \textit{where} \quad I(x) = \{j \mid f_j(x) = f(x), j = 1, 2, ..., n\}$$

The proof of this result can be found in [8]

2.1 Conjugate Functionals

We consider throughout this paper DC programming as:

$$\text{Minimize} \quad f(x) = g(x) - h(x), x \in \mathbb{R}^n \tag{2.1}$$

where g and h are convex functions and $g - h$ is a DC decomposition of f .
The Fenchel conjugate of g is defined as in [12]

$$g^*(y) = \sup\{\langle y, x\rangle - g(x) \mid x \in \mathbb{R}^n\}, y \in \mathbb{R}^n. \tag{2.2}$$

and always convex function.

Definition 2.4. [9] A function $f : \mathbb{R}^n \to \mathbb{R}$ is γ-strongly convex if and only if the function

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

is convex. In particular, if f is strongly convex, then f is also strictly convex.

Theorem 2.5. *[9] Let f be problem in (2.1) and the sequence $\{x_k\}$ generated by the DCA algorithm. Suppose that g and h are γ_1 and γ_2 convex respectively, then*

$$f(x_k) - f(x_{k+1}) \geq \frac{\gamma_1 + \gamma_2}{2}||x_{k+1} - x_k||^2, \forall k \in N. \tag{2.3}$$

Proof. Since $y_k \in \partial h(x_k)$, then one has

$$\langle y_k, x - x_k\rangle + \frac{\gamma_2}{2}||x - x_k||^2 \leq h(x) - h(x_k) \text{for all} x \in \mathbb{R}^n.$$

In particular,h

$$\langle y_k, x_{k+1} - x_k\rangle + \frac{\gamma_2}{2}||x_{k+1} - x_k||^2 \leq h(x_{k+1}) - h(x_k)$$

In addition,$x_{k+1} \in \partial g^*(y_k)$, and so $y_k \in \partial g(x_{k+1})$, which similarly implies

$$\langle y_k, x_k - x_{k+1}\rangle + \frac{\gamma_1}{2}||x_k - x_{k+1}||^2 \leq g(x_k) - g(x_{k+1})$$

Adding these inequalities gives (2.3).

$$\frac{\gamma_1 + \gamma_2}{2}||x_{k+1} - x_k||^2 \leq f(x_k) - f(x_{k+1}), \forall k \in N.$$

□

Theorem 2.6. *[9] Let f be defined by (2.1) and $\{x_k\}$ is a sequence created by DCA. Then functional value $\{f(x_k)\}$ is a decreasing sequence. If f is bounded from below and g is lower semicontinuous, and that g is γ_1-convex and h is γ_2-convex with $\gamma_1 + \gamma_2 > 0$. If $\{x_k\}$ is bounded, then all subsequential limits of $\{x_k\}$ converge to a stationary point of f.*

Definition 2.7. [8] Let F be a nonempty closed subset of $\mathbb{R}^n$ and let $x \in \mathbb{R}^n$.

(i) Define the distance between x and set F by

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

(ii) The set of all Euclidean projection from x to F is defined by

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

It is well-known that $P(x, F)$ is nonempty 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 2.8. *[11, 20] Given any $a \in \mathbb{R}^n$ and $\gamma > 0$, A Nesterov smoothing approximation of $\varphi(x) = ||x - a||_1$ has the representation*

$$\varphi_\gamma(x) := \frac{1}{2\gamma}\|x - a\|^2 - \frac{\gamma}{2}[d(\frac{x - a}{\gamma}; F)]^2.$$

where F is the closed unit box of $\mathbb{R}^n$, i.e., $F := \{v = (v^1, ..., v^n) \in \mathbb{R}^n \mid -1 \leq v_i \leq 1$ for $i = 1, ..., n\}$ Moreover, $\nabla\varphi_\gamma(x) = P(\frac{x-a}{\gamma}; F)$ and

$$\varphi_\gamma(x) \leq \varphi(x) \leq \varphi_\gamma(x)) + \frac{\gamma}{2}\|F\|^2, \tag{2.4}$$

where $\|F\| := \sup\{\|q\| \mid q \in F\}$.

3 Problems Formulation

Clustering is typically a partition of N data points into k groups (clusters) such that the points in each group are more similar to each other than to points in other groups. To define our problems, consider a set A of i data points, that is $A = \{a^i \in R^n : i = 1, ..., m\}$ and k variable cluster centers denoted by $x^1, ..., x^k$. We modeled the clustering problem by choosing k separate centers from the data points that minimize the overall distance. Other members of the data will be assigned to one of the centers based on the ℓ_1- norm between the data points and centers. In the model, nodes are grouped in to k variable centers by minimizing the ℓ_1- norm from all node to the k centers. Then we define clustering model as

$$\text{Minimize} \sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1 \tag{3.1}$$

where the distance measure is the ℓ_1-norm defined as

$$||x||_1 = \sum_{j=1}^{n} |x_j|.$$

In addition, to make sure that the k centers are selected from the existing nodes, we need to add the following constraint to the optimization problem formulated in (3.1):

$$\sum_{j=1}^{k} \min_{i=1,...,m} \|x^j - a^i\|_1 = 0. \tag{3.2}$$

The constraints given in (3.2) helps to push the artificial center to the closest node.
Now we formulate the clustering model as

$$\text{Minimize} \sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1 \tag{3.3}$$

subject to

$$\sum_{j=1}^{k} \min_{i=1,...m} \|x^j - a^i\|_1 = 0, \quad x^1, ..., x^k \in \mathbb{R}^n.$$

To solve the problem given in (3.3) we apply penalty method with parameter $\tau > 0$. Then we can express the minimization problem as unconstrained problem as follows:

$$\text{Minimize} \sum_{i=1}^{m} \min_{j=1,...,k} \|x^j - a^i\|_1 + \tau \sum_{j=1}^{k} \min_{i=1,...,m} \|x^j - a^i\|_1, \quad x^1, ..., x^k \in \mathbb{R}^n$$

Since the pointwise minimum of convex functions may not be convex, we can suitably write as difference of convex functions as given in (3.4).

$$\min_{i=1,...,m} f_i(x) = \sum_{i=1}^{m} f_i(x) - \max_{t=1,...,m} \sum_{i=1, i\neq t}^{m} f_i(x). \tag{3.4}$$

Rewriting the minimum of convex function in (3.4) as sum and maximum of convex functions we have

$$\begin{aligned} f(x^1, x^2, ..., x^k) \quad = \quad & \sum_{i=1}^{m} \sum_{j=1}^{k} \|x^j - a^i\|_1 - \sum_{i=1}^{m} \max_{t=1,...,k} \sum_{j=1, j\neq t}^{k} \|x^j - a^i\|_1 \\ & + \tau \sum_{i=1}^{m} \sum_{j=1}^{k} \|x^j - a^i\|_1 - \tau \max_{t=1,...,k} \sum_{j=1, j\neq t}^{k} \|x^j - a^i\|_1 \end{aligned}$$

Writing as DC function it becomes

$$\begin{aligned} f(x^1, x^2, ..., x^k) &= (1+\tau)\sum_{i=1}^{m}\sum_{j=1}^{k}\|x^j - a^i\|_1 - \sum_{i=1}^{m}\max_{t=1,...,k}\sum_{j=1,j\neq t}^{k}\|x^j - a^i\|_1 \\ &\quad -\tau\sum_{j=1}^{k}\max_{r=1,...,m}\sum_{i=1,i\neq r}^{m}\|x^j - a^i\|_1 \end{aligned} \tag{3.5}$$

where the corresponding convex functions g and h are given as

$$g(x^1, x^2, ..., x^k) = (1+\tau)\sum_{i=1}^{m}\sum_{j=1}^{k}\|x^j - a^i\|_1$$

$$h(x^1, x^2, ..., x^k) = \sum_{i=1}^{m}\max_{t=1,...,k}\sum_{j=1,j\neq t}^{k}\|x^j - a^i\|_1 + \tau\sum_{j=1}^{k}\max_{r=1,...,m}\sum_{i=1,i\neq r}^{m}\|x^j - a^i\|_1$$

Since f is DC function based on ℓ_1 smoothing studied in [20], we give a Nesterov's approximation of $||x-a||_1$ as define in Proposition 2.8 :

$$\|x-a\|_1 = \frac{\gamma}{2}\left[\|\frac{x-a}{\gamma}\|^2 - [d(\frac{x-a}{\gamma}; F)]^2\right]$$

and a Nesterov smoothed objective function in (3.5) is as follows.

$$\begin{aligned} f_\gamma(x^1, x^2, ..., x^k) &= \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}\|\frac{x^j - a^i}{\gamma}\|^2 \\ &\quad -\frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}[d(\frac{x^j - a^i}{\gamma}; F)]^2 \\ &\quad -\sum_{i=1}^{m}\max_{t=1,...,k}\sum_{j=1,j\neq t}^{k}\|x^j - a^i\|_1 - \tau\sum_{j=1}^{k}\max_{r=1,...,m}\sum_{i=1,i\neq r}^{m}\|x^j - a^i\|_1) \end{aligned} \tag{3.6}$$

The aim is to minimize the smoothed problem f_γ as:

$$\text{Minimize} \quad \{f_\gamma(x^1, x^2, ..., x^k) = g_\gamma(x^1, x^2, ..., x^k) - h_\gamma(x^1, x^2, ..., x^k)\}$$

where

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

$$\begin{aligned} h_\gamma(x^1, x^2, ..., x^k) &= \frac{(1+\tau)\gamma}{2}\sum_{i=1}^{m}\sum_{j=1}^{k}[d(\frac{x^j - a^i}{\gamma}; F)]^2 + \sum_{i=1}^{m}\max_{t=1,...,k}\sum_{j=1,j\neq t}^{k}\|x^j - a^i\|_1 \\ &\quad +\tau\sum_{j=1}^{k}\max_{r=1,...,m}\sum_{i=1,i\neq r}^{m}\|x^j - a^i\|_1. \end{aligned}$$

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

$$\|A\|_F = \sqrt{\langle A, A\rangle} = \sqrt{\sum_{j=1}^{k}\langle a^j, a^j\rangle} = \sqrt{\sum_{j=1}^{k}||a^j||^2} \tag{3.7}$$

To solve the smoothed problem, first we compute partial of g_γ to implement DCA using matrix norm defined in (3.7). Then it becomes

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

The partials of g_γ with respect to X can be written as

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

where E_{km} is a matrix with elements are all ones. Then, $g_{1\gamma}$ is smooth and its partial gradients are:

$$\begin{aligned} \nabla_x g_\gamma(x^1, x^2, ..., x^k) &= \frac{(1+\tau)}{\gamma}\left[mX - E_{km}A\right] \\ &= \frac{(1+\tau)}{\gamma}\left[mX - M\right], \text{where} \quad M = E_{km}A. \end{aligned}$$

Next,we focus on $X \in \partial g^*(Y)$ where g^* is a Fenchel conjugate defined in (2.2) and can be calculated using the fact that $X \in \partial g^*(Y) \Leftrightarrow Y \in \partial g(X)$. Since g_γ is differentiable, it is expressed as

$$\nabla_x g_\gamma(x^1, x^2, ..., x^k) = Y.$$

And now one can drive X from the following equation as

$$Y = \frac{(1+\tau)}{\gamma}\left[mX - M\right]$$

Now we can express X as

$$X = \frac{\gamma Y}{m(1+\tau)} + \frac{M}{m}.$$

To compute the subgradient of convex function h_γ in (3.7). First we express h_γ as sum of convex functions as in (3.8)

$$h_\gamma(x^1, x^2, ..., x^k) = h_{1\gamma}(x^1, x^2, ..., x^k) + h_{2\gamma}(x^1, x^2, ..., x^k) + h_{3\gamma}(x^1, x^2, ..., x^k) \tag{3.8}$$

where

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

Since we use ℓ_1 norm, the subgradient $Y \in \partial h_\gamma(X)$ for the case where F is the closed unit box in $\mathbb{R}^n$ defined as $F = \{(v_1, ..., v_n) \in \mathbb{R}^n \mid 1 \leq v_k \leq 1, k = 1, ..., n\}$.
For a given $v \in \mathbb{R}$ we define

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

then subgradient of $f(x) = ||x||_1$ at $x \in \mathbb{R}^n$ is sign(x).
Let us find the gradient of a smooth function h_1 in (3.8) with respect to X. As all function in sum is convex, subgradient of h_γ is computed using subdifferential sum and maximum rule, given in [8]. Hence

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

Thus, the partial of $h_{1\gamma}$ at X and λ is given as:

$$\frac{\partial h_{1\gamma}}{\partial x^j}(x^1, x^2, ..., x^k) = (1+\tau) \sum_{i=1}^{m} \left[\frac{x^j - a^i}{\gamma} - P(\frac{x^j - a^i}{\gamma}; F) \right], \tag{3.9}$$

for all $j = 1, ..., k$. Note that, H_1 is a $k \times n$ matrix.

The projection in (3.9) is the Euclidean projection from $v \in \mathbb{R}^n$ onto a unit closed box F is defined as,

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

Where $e \in \mathbb{R}^n$ is a vector with one in each coordinate.
On the other hand, the third kinds $h_{2\gamma}$ and $h_{3\gamma}$, are non-smooth since we use ℓ_1 norm. Then, we can compute a sub-gradient for $h_{2\gamma}$ and $h_{2\gamma}$, based on the subdifferential formula for maximum functions, as demonstrated below. For instance, we consider

$$\begin{aligned} h_{2\gamma}(x^1, x^2, ..., x^k) &= \sum_{i=1}^{m} \max_{t=1,...,k} \sum_{j=1, j\neq t}^{k} ||x^j - a^i||_1 \\ &= \sum_{i=1}^{m} \max_{t=1,...,k} \left(\sum_{j=1}^{k} ||x^j - a^i||_1 - ||x^t - a^i||_1 \right) \end{aligned}$$

For $i = 1, ..., m$, select an index $t(i)$ so that

$$\max_{t=1,...,k} \left(\sum_{j=1}^{k} ||x^j - a^i||_1 - ||x^t - a^i||_1 \right) = \sum_{j=1}^{k} ||x^j - a^i||_1 - ||x^{t(i)} - a^i||_1$$

Now let $U = (u_{ji})$ be a signed block matrix with $u_{ji} = \text{sign}(x^j - a^i)$ is row vector and U^i be i^{th} column block matrix of U. Then $\partial h_{2\gamma}$ at X is

$$\frac{\partial h_{2\gamma}}{\partial x^j} := \sum_{i=1}^{m} \left(U^i - e_{t(i)} u_{t(i)i} \right)$$

for $e_{t(i)}$ a column vector of k coordinates with one at the $t(i)^{th}$ place and zero otherwise.
Similarly the subgradient of $h_{3\gamma}$ is given by

$$\begin{aligned} h_{3\gamma}(x^1, x^2, ..., x^k) &= \tau \sum_{j=1}^{k} \max_{r=1,...,m} \sum_{i=1, i\neq r}^{m} ||x^j - a^i||_1 \\ &= \sum_{j=1}^{k} \max_{r=1,...,m} \left(\sum_{i=1}^{m} ||x^j - a^i||_1 - ||x^j - a^r||_1 \right) \end{aligned}$$

for all $j = 1, ..., k$, select $r(j)$ so that

$$\max_{r=1,...,m} \left(\sum_{i=1}^{m} ||x^j - a^i||_1 - ||x^j - a^r||_1 \right) = \sum_{i=1}^{m} ||x^j - a^i||_1 - ||x^j - a^{r(i)}||_1.$$

Let $S \in \mathbb{R}^{k \times n}$ be a matrix with j^{th} row $\sum_{i=1}^{m} s_{ji} - s_{jr(j)}$, then $\partial h_2\gamma$ at X is

$$\frac{\partial h_{3\gamma}}{\partial x^j} := \tau S.$$

From the subgradient calculated above we have,

$$Y = \frac{\partial h_{1\gamma}}{\partial x^j} + \frac{\partial h_{2\gamma}}{\partial x^j} + \frac{\partial h_{3\gamma}}{\partial x^j}.$$

Now we have brought all necessary point such as gradient and subgradient to formulate the DCA algorithm that solve the problem as shown in Table 1.

DCA Algorithm
1. Initial: $A, X0, \tau 0, \gamma 0, \sigma_1, \sigma_2, \varepsilon, , N \in \mathbb{N}$.
2. while stopping is not reached do
3. For $k = 1, ..., N$ do
4. Search $Y_k \in \partial h_\gamma(X_{k-1})$
5. Where $X_k = \frac{\gamma Y_k}{m(1+\tau)} + \frac{M}{m}$
12. end for
13. update τ and γ
14. end while
15. Output (X_N)

Table 1. DCA Algorithm for the problem

4 Simulation Results

The numerical simulation was done on a laptop with MATLAB software by considering artificial data. We used a continuous formulation of discrete problem. As a result it became non-smooth and non-convex, on which Nesterov's smoothing and DC-based algorithms were implemented. During numerical simulation different parameters were used, among those, we used a large growing τ. In addition penalty and smoothing parameter are updated throughout the iteration as follows : $\tau_{i+1} = \sigma_1\tau_i, \sigma_1 > 1$, and $\gamma_{i+1} = \sigma_2\gamma_i, \sigma_2 \in (0, 1)$. We selected starting penalty parameter ($\tau_0 = e^{-6}$) and initial smoothing parameter $\gamma_0 = 1$. Besides, we used $\sigma_1 = 16e^9$ as penalty parameter's growth factor and $\sigma_2 = 0.75$ smoothing parameter's decay factor after testing with different values.

For the implementation of the algorithms, we used a randomly selected starting cluster centers. Since the algorithms are modified DCA, there is no guarantee that our algorithms converge to a global optimal solution. However, for small datasets the algorithm converges 100 % of the time to a global optimal solution.

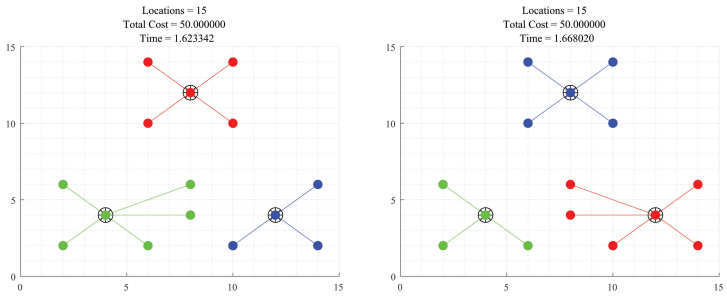

Figure 1. Multiple optimal solution and CPU time of DCA algorithm , with 15 nodes data points and three clusters. The total cost $f(X) = 50.00$, with centers

$$X = \begin{pmatrix} 8.0000 & 12.0000 \\ 4.0000 & 4.0000 \\ 12.0000 & 4.0000 \end{pmatrix}$$

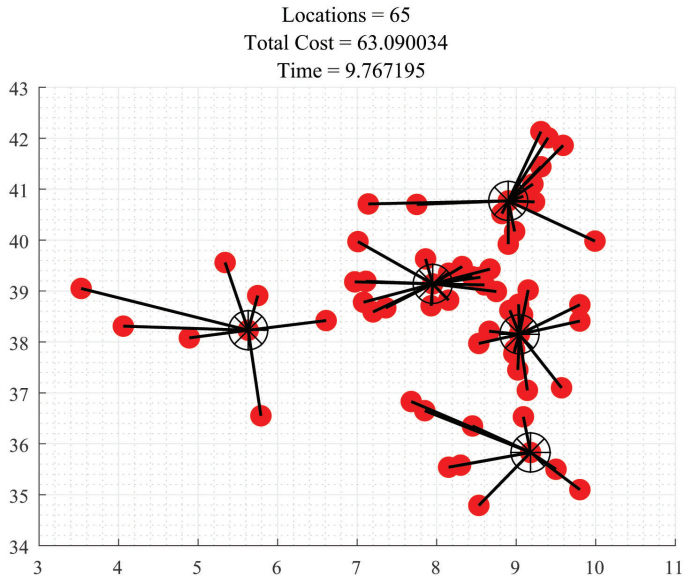

Figure 2. Optimal solution and CPU time of DCA algorithm of 65 town in Oromia region of Ethiopia , with five clusters. The total cost $f(X) = 63.090034$, with centres

$$X = \begin{pmatrix} 9.2100 & 41.1000 \\ 7.9500 & 39.1400 \\ 9.1800 & 35.8300 \\ 9.0400 & 38.1500 \\ 5.6300 & 38.2300 \end{pmatrix}$$

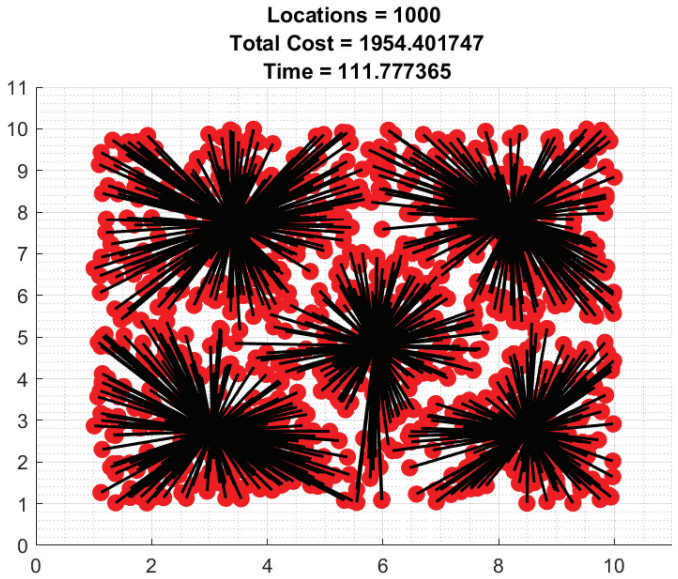

Figure 3. Optimal solution and CPU time of DCA algorithm , with 1000 nodes random data points and five clusters. The total cost $f(X) = 1954.401747$, with centers

$$X = \begin{pmatrix} 8.2370 & 7.8794 \\ 3.0467 & 2.7415 \\ 3.4140 & 7.7048 \\ 8.5211 & 2.7217 \\ 5.8614 & 4.7893 \end{pmatrix}$$

5 Conclusion

In this paper ,we formulate a clustering problem as continuous optimization using DC functions where the distance between two data points is measured by ℓ_1 norm. We implemented a DC based algorithms and tested with real and artificial dataset of various sizes and dimensions with MATLAB. Starting with different random initial cluster centers the algorithm converge to optimal value in reasonable time. As a result an improved iteration time for large scale problems and convergence to optimal cluster centers were observed. In addition the algorithm used in solving clustering problems with DCA is flexible to apply to other nonsmooth nonconvex optimization problem such as location science and machine learning.

References

[1] L. T. H. An, M. T. Belghiti, P. D. Tao: A new efficient algorithm based on DC programming and DCA for clustering. J. Glob. Optim., **27**,(2007), 503–608.

[2] An, L.T.H., Minh, L.H., Tao, P.D., New and efficient DCA based algorithms for minimum sum-of-squares clustering. Pattern Recognit, Ann. New York Acad. Sci., **47**,(2014),388–401.

[3] L. T. H. An, and L. H. Minh, Optimization based DC programming and DCA for hierarchical clustering. *European J. Oper. Res*. **183** (2007), 1067–1085.

[4] L. T. H. An, and P. D. Tao, Convex analysis approach to D.C. programming: Theory, algorithms and applications. *Acta Math. Vietnam.* **22** (1997), 289–355.

[5] A. Bagirov, Long Jia, I. Ouveysi, and A.M. Rubinov, Optimization based clustering algorithms in Multicast group hierarchies, in: Proceedings of the Australian Telecommunications, Networks and Applications Conference (ATNAC), (2003), Melbourne Australia (published on CD, ISNB 0-646-42229-4).

[6] Bagirov, A., Taheri, S., Ugon, J., Nonsmooth DC programming approach to the minimum sum-of-squares clustering problems , Pattern Recognit, **53**,(2016), 12–24.

[7] Barbosa, G. V., Villas-Boas, S. B., Xavier, A. E., Solving the two-level clustering problem by hyperbolic smoothing approach, and design of multicast networks, In: Selected Proceedings, WCTR RIO, 2013

[8] B. S. Mordukhovich and N. M. Nam, *An Easy Path to Convex Analysis and Applications*, Morgan & Claypool Publishers, San Rafael, CA, 2014.

[9] Nam, N.M., Rector, R.B., Giles, D., Minimizing differences of convex functions with applications to facility location and clustering , J. Optim. Theory Appl.,**173**,(2017) ,255–278.

[10] Nam N. M., Geremew, W., Reynolds, R., & Tran, T. (2018). *Nesterov's smoothing technique and minimizing differences of convex functions for hierarchical clustering.* Optimization Letters. **12**(3), 455–473.

[11] Y. Nesterov: Smooth minimization of non-smooth functions. Math. Program. **103**, 127–152 (2005).

[12] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[13] T. Pham Dinh and H. A. Le Thi, A d.c. optimization algorithm for solving the trust-region subproblem, SIAM J. Optim.,**8**, (1998), 476–505.

[14] Geremew, W., Nam, N.M., Semenov, A., Boginski, V., & Pasiliao, E. (2018). *A DC programming approach for solving multicast network design problems via the Nesterov smoothing technique.* Journal of Global Optimization. **72**(4), 705–729.

[15] N. M. Nam, R.B. Rector, D. Giles: Minimizing Differences of Convex Functions with Applications to Facility Location and Clustering, Math. Program, **173**, (2017), 255–278.

[16] H.A. LE THI AND T. PHAM DINH, DC Programming and DCA: Thirty Years of, Math. Program.,**169**, (2018) , 5–68.

[17] Anuj Bajaj, Boris S. Mordukhovich, Nguyen Mau Nam, and Tuyen Tran., Solving Multifacility Location Problems by DC Algorithms, J. Optim., 2019.

[18] Yu. Nesterov, Lectures on Convex Optimization, 2nd edition, Springer, Cham, Switzerland, 2018

[19] Nguyen, P.A., Le Thi, H.A, DCA approaches for simultaneous wireless information power transfer in MISO secrecy channel., Optim Eng , 2020.

[20] Mau Nam Nguyen, Hoai An Le Thi, Giles Daniel, Thai An Nguyen, Smoothing techniques and difference of convex functions algorithms for image reconstructions, Optimization,(2019), 1–33.

Author information

Adugna Fita, Department of Applied Mathematics, Adama Science and Technology University, Ethiopia.
E-mail: adugna.fita@astu.edu.et

Wondi Geremew, School of Business, Stockton University, USA.
E-mail: wondi.geremew @stockton.edu

Legesse Lemecha, Department of Applied Mathematics, Adama Science and Technology University, Ethiopia.
E-mail: legesse.lemecha@astu.edu.et

Received: June 17, 2021
Accepted: July 9, 2021