May 4th, 12:00 PM - 2:00 PM

The DC Algorithm & The Constrained Fermat-Torricelli Problem

Nathan Peron Lawrence
Portland State University

George Blikas
Portland State University

The DC Algorithm & The Constrained Fermat-Torricelli Problem

Nathan L., George B.
Faculty Advisor: Dr. Mau Nam Nguyen
Fariborz Maseeh Department of Mathematics and Statistics
PO Box 751, Portland, Oregon 97207

Portland State UNIVERSITY

Abstract

The theory of functions expressible as the Difference of Convex (DC) functions has led to the development of a rich field in applied mathematics known as DC Programming. We survey the work of Pham Dinh Tao and Le Thi Hoai An in order to understand the DC Algorithm (DCA) and its use in solving clustering problems. Further, we present several other methods that generalize the DCA for any norm. These powerful tools enable researchers to reformulate objective functions, not necessarily convex, into DC Programs.

The Fermat-Torricelli problem is visited in light of convex analysis and various norms. Pierre de Fermat proposed a problem in the 17th century that sparked interest in the location sciences: given three points in the plane, find a point such that the sum of its Euclidean distances to the three points is minimal. This problem was solved by Evangelista Torricelli, and is now referred to as the *Fermat-Torricelli problem*. We present the constrained version of the problem using the distance penalty method.

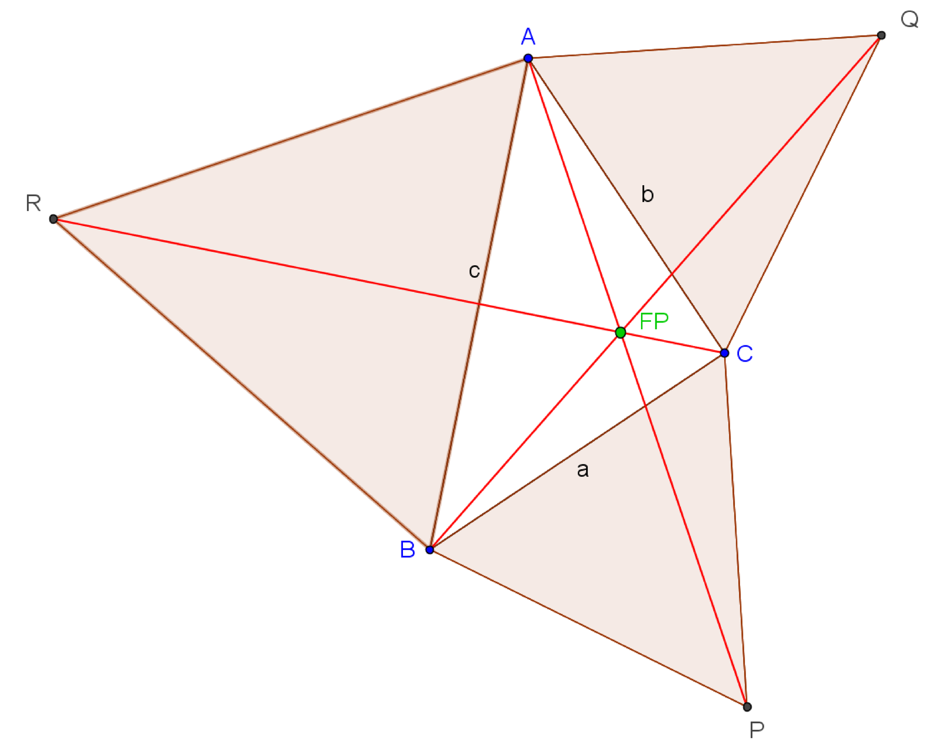

Figure 1: A visualization of the classical Fermat-Torricelli Problem.

The DCA

We start out with an objective function in terms of its DC decomposition:

$$\min_{x\in\mathbb{R}^n} f(x) := g(x) - h(x)$$

The global optimality of our objective function is equivalent to the minimization of its dual, expressed in terms of its Fenchel conjugates:

$$\min_{y\in\text{dom}h^*} h^*(y) - g^*(y)$$

In order to solve this problem, we apply the following algorithm (DCA):

Initialize $x_0 \in \mathbb{R}^n$, $N \in \mathbb{N}$
for k **do** = 1, 2, ... , N
Find $y_k \in \partial h(x_k)$
Find $x_{k+1} \in \partial g^*(y_k)$
end for
Output: x_{N+1}

The DCA will be applied to multifacility location problems.

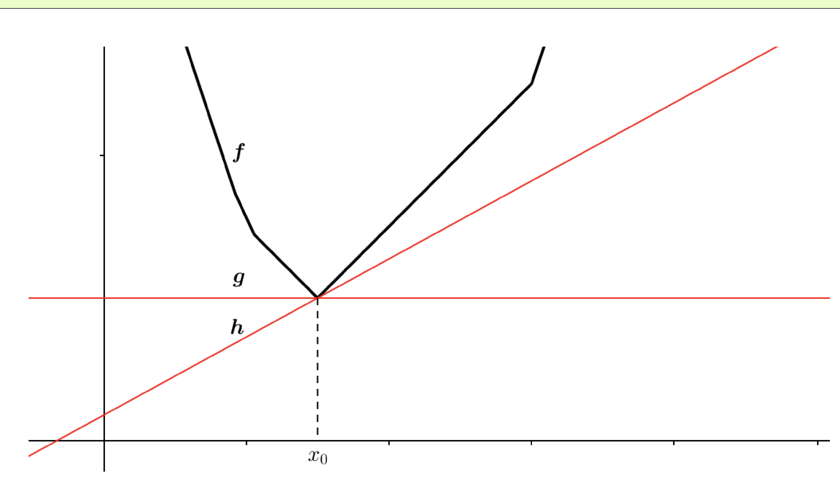

Figure 2: A generalization of gradient on a non-differentiable function. The *subgradient* of a function is denoted $\partial f(\bar{x})$. Note that if $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$ then the function is differentiable

Formulation of the Fermat-Torricelli Problem

The solution to the Fermat-Torricelli Problem is well known. The geometric representation of the classical problem given by Fermat is given in *fig 1*. Given three points in the plane A, B, C, we construct triangle, ABC. If any of the angles $\angle A$, $\angle B$, or $\angle C$ measures 120° or greater, then the respective point is the solution. In the cases where $\angle A$, $\angle B$, and $\angle C$ are less than 120°, on each of sides in $\triangle ABC$ we attach an equilateral triangle whose side lengths are that of the corresponding one. Drawing lines as in *fig 1*, we obtain the desired solution. For a more detailed description, see this link.

This problem has been generalized to any finite number of points in the plane. Harold Kuhn asserted and proved the necessary conditions for generalization in 1972. The Weiszfeld algorithm was the first numerical approach to solve the general case. In the following panels we present the minimizer for the classical problem, and a detailed derivation for that of the constrained problem.

The Fermat-Torricelli Problem and The Weiszfeld Algorithm

Given multiple points in the plane and an initial starting point, the Weiszfeld Algorithm is a numerical method for obtaining the optimal center. We present the Weiszfeld Algorithm. We denote the set of data points $\{a_1, a_2, ..., a_m\} \subset \mathbb{R}^n$, and initial vector $x_0 \in \mathbb{R}^n$.

Given $\{a_1, a_2, ..., a_m\} \subset \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$ and $N \in \mathbb{N}$
for k = 0, 1, 2, ..., N **do**
$x_{k+1} = \dfrac{\sum_{i=1}^m \frac{a_i}{\|x-a_i\|}}{\sum_{i=1}^m \frac{1}{\|x_k-a_i\|}}$
end for
Output: x_{N+1}

The formal conditions for convergence were not shown before the algorithm's development. In 1972, Harold Khun showed that given the sequence $\{x_k\}_k$ generated by the Weiszfeld Algorithm, $x_k \to \bar{x}$ precisely when $x_k \notin \{a_1, a_2, ..., a_m\}$.

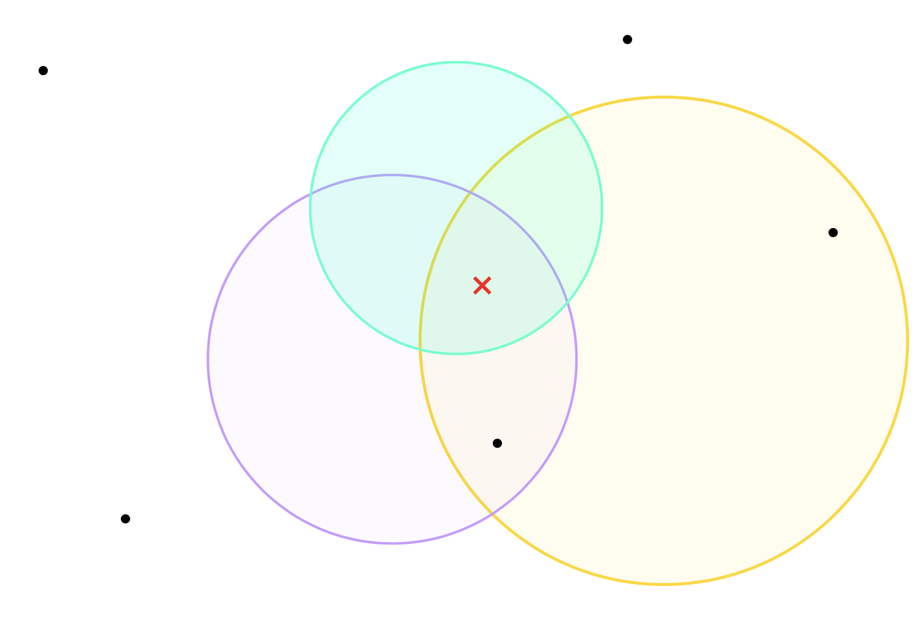

Figure 3: The three circles represent convex subsets of $\mathbb{R}^2$. Similar to the original problem, but we are looking for a point in the shaded region whose sum of distances to each black point is minimal with respect to the shaded region.

The Constrained Fermat-Torricelli Problem

Up until this point, we have only considered $x \in \mathbb{R}^n$; that is, a single constraint on the objective function. Next, we consider a finite sequence of convex subsets $\{\Omega_i\}_{i=1}^N$ of $\mathbb{R}^n$. Given data points $\{a_1, a_2, ..., a_m\} \subset \mathbb{R}^n$, the constrained Fermat-Torricelli problem becomes the following:

$$\text{minimize} \quad f(x) = \sum_{i=1}^m ||x - a_i|| \tag{1}$$
$$\text{subject to} \quad x \in \bigcap_{i=1}^N \Omega_i$$

In a similar fashion of the algorithm presented above, the following is a minimizer for (1):

$$x = \frac{\sum_{i=1}^m \frac{a_i}{\|x-a_i\|} + 2\mu \sum_{i=1}^p P(x;\Omega_i)}{\sum_{i=1}^m \frac{1}{\|x-a_i\|} + 2\mu p},$$

where $\mu > 0$ is a penalty constant. We present the associated numerical method for obtaining the minimizer, $\bar{x}$. Define

$$F(x) := x, \quad x \notin \{a_1, a_2, ..., a_m\}$$

Given $\{a_1, a_2, ..., a_m\} \subset \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$ and $N \in \mathbb{N}$
for k = 0, 1, 2, ..., N **do**
$x_{k+1} = F(x_k)$
end for
Output: x_{N+1}

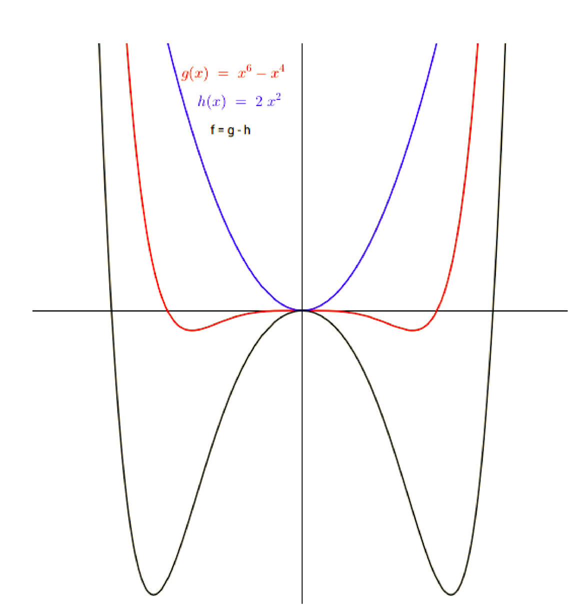

Figure 4: The function f is an example of a DC function.

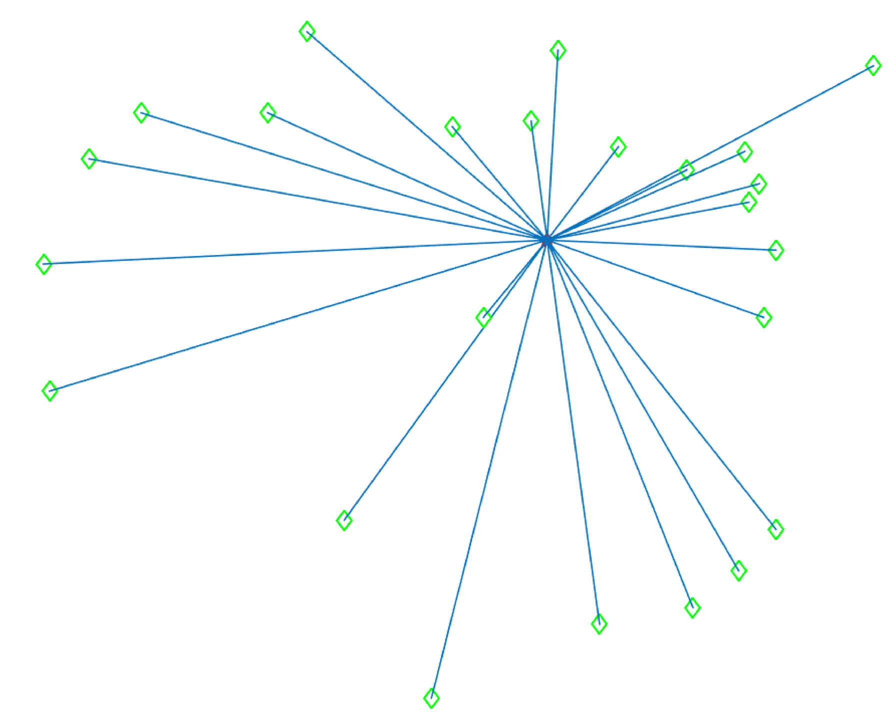

Figure 5: After choosing an initial starting point, we iteratively minimize the Euclidean distance to all neighbouring points (diamonds). Where the lines converge is the optimal solution.

Some Motivation for Multifacility Location Problems

In the previous algorithms, we have only considered one case of a more general framework. It is often the case that in the world of commerce multiple facilities are needed to export and deliver product to a wide range of consumers, who may live in wide-spanning locations. In order to determine the optimal location for facility development, industry turns to some the numerical methods we present. We apply the DCA, in combination with Nesterov's Smoothing Technique to these problems.

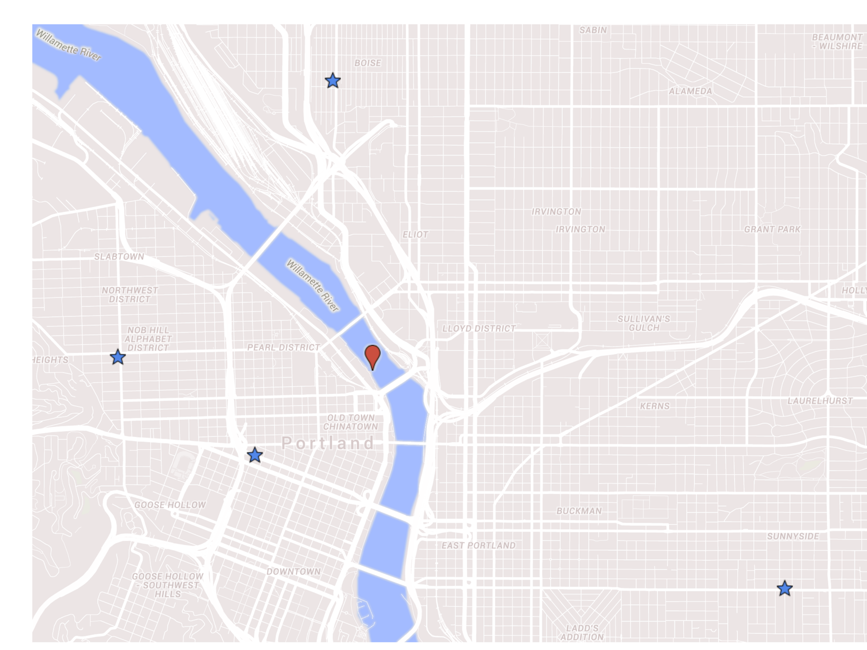

Figure 6: The above map demonstrates how the optimal solution isn't always feasible in application

Location Analysis with Generalized Distances

Using Nesterov's Smoothing Technique and the DCA we are able to solve multifacility location problems under different norms. As a corollary, this allows us to further generalize the Fermat-Torricelli problem in terms of a *Minkowski gauge*. For our purposes, let F be the closed unit ball in $\mathbb{R}^n$, then the Minkowski gauge associated with F is defined as $\rho_F(x) := \|x\|$. Further, $F^\circ := \{y \in \mathbb{R}^n : \langle x, y\rangle \le 1 \text{ for all } x \in F\}$.

Let $\{a_1, a_2, ..., a_m\}$ be our set of data points in $\mathbb{R}^n$. For k centroids we want to minimize the objective function given by

$$f(x_1, \ldots, x_k) := \sum_{i=1}^m \min_{\ell=1,\ldots,k} \rho_F(x_\ell - a_i), \text{ where } x_\ell \in \mathbb{R}^n \text{ for } \ell = 1, \ldots, k. \tag{2}$$

Despite the objective function f not necessarily being convex, the problem given by (2) always admits an optimal solution in $(\mathbb{R}^n)^k$. We can approximate f as a DC program, denoted $f_\mu(x) = g_\mu(x) = h_\mu(x)$ where $x \in (\mathbb{R}^n)^k$ and $\mu > 0$ is the smoothing parameter. Specifically, we have

$$g_\mu(x_1, \ldots, x_k) := \frac{1}{2\mu} \sum_{i=1}^m \sum_{\ell=1}^k \|x_\ell - a_i\|$$
$$h_\mu(x_1, \ldots, x_k) := \frac{\mu}{2} \sum_{i=1}^m \sum_{\ell=1}^k \left[d(\frac{x_\ell - a_i}{\mu}; F^\circ\right]^2 + \sum_{i=1}^m \max_{r=1,\ldots,k} \sum_{\ell=1,\ell\neq r}^k \rho_F(x_\ell - a_i).$$

In the next panel we outline the steps necessary for solving the multifaciltiy location problem with the DCA and the smooth DC decomposition given in this section.

The DCA For Multifacility Location

Given a set of data points $\{a_1, \ldots, a_m\}$ in $\mathbb{R}^n$, let B be the $k \times n$ matrix whose rows are $\sum_{i=1}^m a_i$. With reference to the previous panel, we have the following:

$$\nabla G_\mu(Y) = \frac{1}{m}(B + \mu Y), \text{ where } G_\mu(X) := g_\mu(x_1, \ldots, x_k)$$

Similarly, $H_\mu(X) := h_\mu(x_1, \ldots, x_k) = H_\mu^1 + H_\mu^2$, where H_1, H_2 are the respective summands previously shown. The calculations for their subgradients can be found in [3]

Given $X_1 \in \mathbb{R}^{k\times n}, N \in \mathbb{N}, F, \mu > 0,$ and $a_1, \ldots, a_m \in \mathbb{R}^n$
for k = 1,. . . , N **do**
Calculate $U_k := \nabla H_\mu^1(X_k)$
Find $V_k \in \partial H_\mu^2(X_k)$
Set $Y_k = U_k + V_k$
Let $X_{k+1} = \frac{1}{m}(B + \mu Y_k)$
end for
Output: X_{N+1}.

Results

In this table we compare three algorithms' run-time performance in assigning a single center to data points: the Weiszfeld Algorithm, Algorithm 4, and Algorithm 5 as outlined in Tao, An. The initial center (X) was identical in all trials. The data points (A) were consistent across all algorithms; A was a $n \times 2$ matrix, whose rows represented the data points. Each trial was given 1000 iterations (k). We are running on a 64-bit, 16.00-GB RAM, AMD FX Eight Core 4.00-GHz processor, with Windows 10 OS.

size A	5	50	500	1200
Weiszfeld	0.064774	0.304614	2.749730	6.458274
Algorithm 4	0.078024	0.157531	0.935655	2.154189
Algorithm 5	0.017729	1.253552	121.958931	701.684650

Table 1: Run-time performance in seconds

It is a work in progress to compare the average distances between the various algorithms. In algorithms 4, and 5 there are parameters which by adjusting effects the rate of convergence. We do note, however, that with a decent approximation of those parameters, the average distance from the optimal center is roughly equal across these algorithms. We present some of these preliminary calculations.

size A (n)	5	50	500	1200
Weiszfeld	3.0902e+05	3.0891e+06	3.0894e+07	7.4144e+07
Algorithm 4	2.9402e+05	2.8891e+06	2.8394e+07	6.6944e+07
Algorithm 5	2.5902e+05	2.4891e+06	2.3894e+07	5.4944e+07

Table 2: Distance to centroid; Euclidean distance relative to data points.

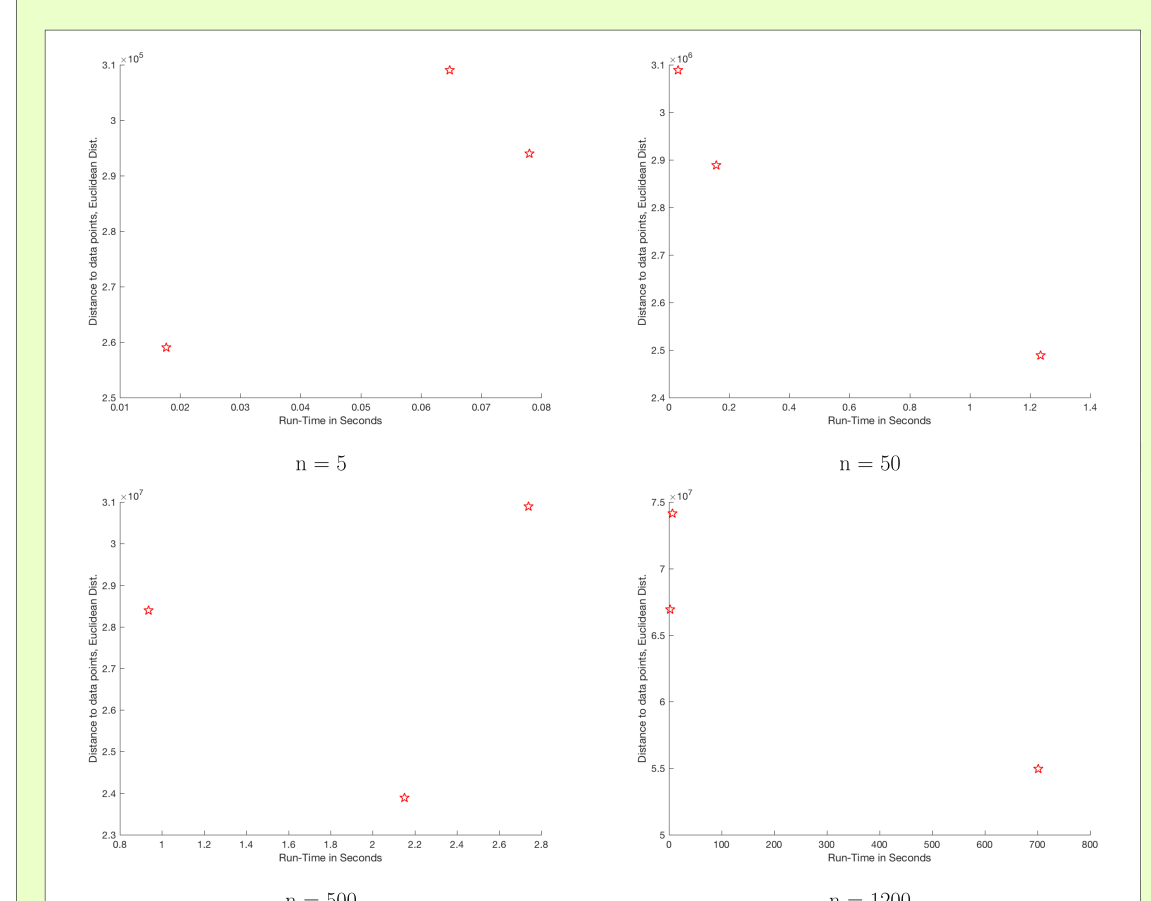

Figure 7: Comparision between the Weiszfeld Algorithm, Algorithm 4 and Algorithm 5, with size of A as 5×2, 50×2, 500×2 and 1200×2, respectively.

Discussion

The three algorithms all find reasonable approximations to the optimal solution. As is documented, for all data sets the total distance to centroid found by algorithm 5 outperformed the others. This is to be expected. In long term calculation the run-time of algorithm 5 is not unreasonable. However, for applications, its run-time performance is unfortunate.

As previously mentioned, it will be our aim in the future to improve upon parameter selection in order to expediate convergence in algorithms 4 and 5. It is clear that algorithm 4 is the best trade-off between run-time and accuracy. A careful selection in parameter choice may make it optimal for application.

References

[1] S. Boyd. *Subgradient Methods.* Notes for EE364b, Stanford University (2014)

[2] B. S. Mordukhovich and N. M. Nam. *An easy path to convex analysis and applications.*Synthesis Lectures on Mathematics and Statistics, 6(2):1218, 2013.

[3] N.M Nam, Rector, R. B., & Giles, D. (2015). *Minimizing Differences of Convex Functions and Applications to Facility Location and Clustering.* arXiv preprint arXiv:1511.07595.

[4] Y. Nesterov. *Smooth minimization of non-smooth functions.* Mathematical Programming Ser. A **103**, 127-152 (2005)

This research was partially supported by the USA National Science Foundation under grant DMS-1411817.