INFORMS Journal on Computing
Vol. 21, No. 4, Fall 2009, pp. 614–629
ISSN 1091-9856 | EISSN 1526-5528 | 09 | 2104 | 0614

DOI 10.1287/ijoc.1080.0315

An Algorithm and a Core Set Result for the Weighted Euclidean One-Center Problem

Piyush Kumar
Department of Computer Science, Florida State University, Tallahassee, Florida 32306,
piyush@cs.fsu.edu

E. Alper Yıldırım
Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey,
yildirim@bilkent.edu.tr

Given a set $\mathcal{A}$ of m points in n-dimensional space with corresponding positive weights, the weighted Euclidean one-center problem, which is a generalization of the minimum enclosing ball problem, involves the computation of a point $c_{\mathcal{A}} \in \mathbb{R}^n$ that minimizes the maximum weighted Euclidean distance from $c_{\mathcal{A}}$ to each point in $\mathcal{A}$. In this paper, given $\epsilon > 0$, we propose and analyze an algorithm that computes a $(1+\epsilon)$-approximate solution to the weighted Euclidean one-center problem. Our algorithm explicitly constructs a small subset $\mathcal{X} \subseteq \mathcal{A}$, called an *$\epsilon$-core set* of $\mathcal{A}$, for which the optimal solution of the corresponding weighted Euclidean one-center problem is a close approximation to that of $\mathcal{A}$. In addition, we establish that $|\mathcal{X}|$ depends only on ϵ and on the ratio of the smallest and largest weights, but is independent of the number of points m and the dimension n. This result subsumes and generalizes the previously known core set results for the minimum enclosing ball problem. Our algorithm computes a $(1+\epsilon)$-approximate solution to the weighted Euclidean one-center problem for $\mathcal{A}$ in $\mathcal{O}(mn|\mathcal{X}|)$ arithmetic operations. Our computational results indicate that the size of the ϵ-core set computed by the algorithm is, in general, significantly smaller than the theoretical worst-case estimate, which contributes to the efficiency of the algorithm, especially for large-scale instances. We shed some light on the possible reasons for this discrepancy between the theoretical estimate and the practical performance.

Key words: weighted Euclidean one-center problem; minimum enclosing balls; core sets; approximation algorithms
History: Accepted by John Hooker, Area Editor for Constraint Programming and Optimization; received February 2008; revised September 2008; accepted November 2008. Published online in *Articles in Advance* April 7, 2009.

1. Introduction

Given a finite set of points $\mathcal{A} := \{a^1, \ldots, a^m\} \subset \mathbb{R}^n$ with corresponding positive weights $\mathcal{W} := \{\omega_1, \ldots, \omega_m\}$, the weighted Euclidean one-center problem is concerned with finding the point $c_{\mathcal{A}} \in \mathbb{R}^n$ that minimizes the maximum weighted Euclidean distance from $c_{\mathcal{A}}$ to each point in $\mathcal{A}$. Formally, it amounts to solving the following optimization problem:

$$\rho_{\mathcal{A}} := \min_{c \in \mathbb{R}^n} \max_{i=1,\ldots,m} \omega_i \|a^i - c\|.$$

The weighted Euclidean one-center problem reduces to the minimum enclosing ball (or the Euclidean one-center) problem when all the weights are identical. It follows that $c_{\mathcal{A}}$ and $\rho_{\mathcal{A}}$ are simply the center and the radius of the minimum enclosing ball of $\mathcal{A}$, respectively, if all weights ω_i are equal to one. Henceforth, we use $(\mathcal{A}, \mathcal{W})$ to denote an instance of the weighted Euclidean one-center problem.

The weights ω_i can be viewed as a measure of importance of the input point a^i. More precisely, input points with larger weights have a higher tendency to "attract" the optimal center towards themselves in comparison with points with smaller weights. As such, the weighted Euclidean one-center problem has extensive applications in facility location (Drezner and Gavish 1985). Typically, the objective is to minimize the maximum weighted response time as in the examples of emergency services, health care, and firefighting, or to minimize the maximum weighted travel time as in the examples of post offices, warehouses, and schools.

For $c \in \mathbb{R}^n$, let

$$\rho(c) := \max_{i=1,\ldots,m} \omega_i \|a^i - c\|. \tag{1}$$

Given $\epsilon > 0$, we say that $(c, \rho(c)) \in \mathbb{R}^n \times \mathbb{R}$ is a $(1+\epsilon)$-approximate solution to the weighted Euclidean one-center problem for the instance $(\mathcal{A}, \mathcal{W})$ if

$$\rho_{\mathcal{A}} \le \rho(c) \le (1+\epsilon)\rho_{\mathcal{A}}. \tag{2}$$

A subset $\mathcal{X} \subseteq \mathcal{A}$ is said to be an ϵ-core set (or a core set) of $\mathcal{A}$ if

$$\rho_{\mathcal{X}} \le \rho_{\mathcal{A}} \le (1+\epsilon)\rho_{\mathcal{X}}, \tag{3}$$

where $(c_{\mathcal{X}}, \rho_{\mathcal{X}}) \in \mathbb{R}^n \times \mathbb{R}$ denotes the optimal solution of the weighted Euclidean one-center problem of the instance $(\mathcal{X}, \{\omega_j : a^j \in \mathcal{X}\})$. Since $c_{\mathcal{A}}$ lies in the convex hull of $\mathcal{A}$ (see §2), it follows that there always exists a 0-core set of size at most $n+1$.

Small core sets provide a compact representation of a given instance of an optimization problem. Furthermore, the existence of small core sets paves the way for the design of efficient algorithms, especially for large-scale instances. Recently, several approximation algorithms have been developed for various classes of geometric optimization problems based on the existence of small core sets (Bădoiu et al. 2002, Kumar et al. 2003, Bădoiu and Clarkson 2003, Tsang et al. 2005, Kumar and Yıldırım 2005, Agarwal et al. 2005, Yıldırım 2008, Todd and Yıldırım 2007). Computational experience indicates that such algorithms are especially well suited for large-scale instances for which a moderately small accuracy (e.g., $\epsilon = 10^{-3}$) suffices.

The weighted Euclidean one-center problem and its variants have been the center of study of many papers (Francis 1967; Hearn and Vijay 1982; Chandrasekaran 1982; Megiddo 1983, 1989; Hansen et al. 1985; Drezner and Gavish 1985; Dyer 1986). In particular, the problem can be solved in time proportional to the number of points for fixed dimension ($n = \mathcal{O}(1)$) (Dyer 1986, Megiddo 1989). However, the dependence on the dimension is exponential. For the case when the dimension is not fixed, Drezner and Gavish (1985) proposed a variant of the ellipsoid method that computes a $(1+\epsilon)$-approximate solution in $\mathcal{O}(n^3 m \log(1/\epsilon))$ arithmetic operations. Incidentally, this asymptotic complexity bound matches with that arising from the application of the ellipsoid method to approximately solve the problem (Grötschel et al. 1988). Because the problem can be formulated as an instance of second-order cone programming, interior-point methods can be applied to compute a $(1+\epsilon)$-approximate solution in polynomial time. However, the cost per iteration becomes prohibitively high as the size of the problem instance increases. We refer the reader to the computational results reported in Zhou et al. (2005) for the special case of the minimum enclosing ball problem.

In this paper, we focus on computing a $(1+\epsilon)$-approximate solution for large-scale instances of the weighted Euclidean one-center problem. Our algorithm explicitly constructs an ϵ-core set $\mathcal{X}$ of $\mathcal{A}$ such that $|\mathcal{X}| = \mathcal{O}(1/(\tau\epsilon))$, where τ is the squared ratio of the minimum weight to the maximum weight. The asymptotic bound on the core set size reduces to $\mathcal{O}(1/\epsilon)$ for the special case of the minimum enclosing ball problem, which matches the previously known core set results (Bădoiu and Clarkson 2003, Kumar et al. 2003, Yıldırım 2008). It has also been shown that this bound is worst-case optimal (Bădoiu and Clarkson 2008). We establish that our algorithm computes a $(1+\epsilon)$-approximate solution in $\mathcal{O}(mn|\mathcal{X}|)$ arithmetic operations. Our extensive computational results indicate that the practical performance of our algorithm is usually much better than that predicted by the worst-case theoretical estimate. We provide some insights into the reasons for this discrepancy between the theoretical estimate and the practical performance.

Our complexity bounds hold in the real number model of computation (Blum et al. 1989). Therefore, the overall complexity bound of our algorithm and the asymptotic bound on the core set are polynomial in the input size for fixed ϵ and τ. However, both of these bounds can actually be expressed in terms of another parameter μ^* (see Corollary 5.1) that arises from our algorithm, which can a priori be bounded below by τ. Our computational results indicate that μ^* behaves like a constant for randomly generated instances even though τ can be arbitrarily small. Therefore, the running time of our algorithm and the size of the resulting core set seem to have a very weak dependence on τ in practice at least for randomly generated instances.

This paper is organized as follows. In the remainder of this section, we define our notation. In §2, we discuss optimization formulations for the weighted Euclidean one-center problem. Section 3 describes a constant factor approximation for our problem. Section 4 gives a simple proof that shows the existence of a core set for our problem. Section 5 is devoted to the presentation and the analysis of our algorithm. We also compare our results to other related results in the literature in this section. The computational results are presented in §6. Finally, §7 concludes the paper.

1.1. Notation

Vectors are denoted by lowercase roman letters. For a vector p, p_i denotes its ith component. Inequalities on vectors apply to each component. We reserve e^j for the jth unit vector, e for the vector of all ones, and I for the identity matrix in the appropriate dimension, which will always be clear from the context. Uppercase roman letters are reserved for matrices and M_{ij} denotes the (i, j) component of the matrix M. We use $\log(\cdot)$ and $\log_2(\cdot)$ to denote the natural and the base-2 logarithm, respectively. Functions and operators are denoted by uppercase Greek letters. Scalars except for m and n are represented by lowercase Greek letters unless they represent components of a vector or elements of a sequence of scalars, vectors, or matrices. We reserve i, j, and k for such indexing purposes. Uppercase script letters are used for all other objects such as sets and balls.

2. Optimization Formulations

The weighted Euclidean one-center problem for the instance $(\mathcal{A}, \mathcal{W})$ admits the following formulation as an optimization problem:

$$(\mathcal{P}_1) \quad \min_{c,\rho} \ \rho$$
$$\text{subject to} \quad \omega_i \|a^i - c\| \le \rho, \quad i = 1, \ldots, m,$$

where $c \in \mathbb{R}^n$ and $\rho \in \mathbb{R}$ are the decision variables. By squaring the constraints and defining $\gamma := \rho^2$, $(\mathcal{P}_1)$ can be converted into the following optimization problem with smooth, convex quadratic constraints:

$$(\mathcal{P}_2) \quad \min_{c,\gamma} \ \gamma$$
$$\text{subject to} \quad \nu_i((a^i)^T a^i - 2(a^i)^T c + c^T c) \le \gamma, \quad i = 1, \ldots, m,$$

where

$$\nu_i := \omega_i^2, \quad i = 1, \ldots, m. \tag{4}$$

The Lagrangian dual of $(\mathcal{P}_2)$ is given by

$$(\mathcal{D}) \quad \max_u \ \Phi(u) := \sum_{i=1}^m u_i \nu_i (a^i)^T a^i - \frac{1}{\sum_{i=1}^m u_i \nu_i} \left(\sum_{i=1}^m u_i \nu_i a^i\right)^T \left(\sum_{i=1}^m u_i \nu_i a^i\right)$$
$$\text{subject to} \quad \sum_{i=1}^m u_i = 1, \quad u_i \ge 0, \ i = 1, \ldots, m,$$

where $u \in \mathbb{R}^m$ is the decision variable. It is easy to verify that $(\mathcal{D})$ reduces to the dual formulation of the minimum enclosing ball problem if all the weights are identical (Yıldırım 2008). In contrast with the minimum enclosing ball problem, the objective function of $(\mathcal{D})$ is no longer quadratic for the general weighted problem. We discuss the implications of this observation in further detail in §5.2.

By the Karush-Kuhn-Tucker optimality conditions, $(c_{\mathcal{A}}, \gamma_{\mathcal{A}}) \in \mathbb{R}^n \times \mathbb{R}$ is an optimal solution of $(\mathcal{P}_2)$ if and only if there exists $u^* \in \mathbb{R}^m$ such that

$$\sum_{i=1}^m u_i^* = 1, \tag{5a}$$

$$c_{\mathcal{A}} = \frac{1}{\sum_{i=1}^m u_i^* \nu_i} \sum_{i=1}^m u_i^* \nu_i a^i, \tag{5b}$$

$$\nu_i((a^i)^T a^i - 2(a^i)^T c_{\mathcal{A}} + (c_{\mathcal{A}})^T c_{\mathcal{A}}) \le \gamma_{\mathcal{A}}, \quad i = 1, \ldots, m, \tag{5c}$$

$$u_i^*[\nu_i((a^i)^T a^i - 2(a^i)^T c_{\mathcal{A}} + (c_{\mathcal{A}})^T c_{\mathcal{A}}) - \gamma_{\mathcal{A}}] = 0, \quad i = 1, \ldots, m, \tag{5d}$$

$$u^* \ge 0. \tag{5e}$$

A simple manipulation of the optimality conditions reveals that

$$\gamma_{\mathcal{A}} = \Phi(u^*), \tag{6}$$

which implies that $u^* \in \mathbb{R}^m$ is an optimal solution of $(\mathcal{D})$ and that strong duality holds between $(\mathcal{P}_2)$ and $(\mathcal{D})$. Note that the weighted center $c_{\mathcal{A}}$ of $\mathcal{A}$ is given by a convex combination of the points in $\mathcal{A}$ by (5b).

The existence of the weighted Euclidean one-center of $\mathcal{A}$ directly follows from the maximization of a continuous function over a compact domain in the dual formulation. It is also straightforward to establish the uniqueness by the following simple contradiction argument: If there were two such weighted centers, one could improve the solution by considering an appropriate convex combination of these two centers.

It follows from the optimality conditions that the solution of the weighted Euclidean one-center problem can be obtained by solving the dual problem $(\mathcal{D})$. If $u^* \in \mathbb{R}^m$ denotes an optimal solution of $(\mathcal{D})$, the optimal solution $(c_{\mathcal{A}}, \rho_{\mathcal{A}})$ of $(\mathcal{P}_1)$ is given by

$$c_{\mathcal{A}} = \frac{1}{\sum_{i=1}^m u_i^* \nu_i} \sum_{i=1}^m u_i^* \nu_i a^i, \qquad \rho_{\mathcal{A}} = (\gamma_{\mathcal{A}})^{1/2} = (\Phi(u^*))^{1/2}. \tag{7}$$

By lifting the decision variable to one higher dimension, Dyer (1986) proposes an alternative optimization formulation of the weighted Euclidean one-center problem with m linear constraints and one convex quadratic constraint. However, the feasible region of the resulting dual problem is different from that of our dual problem $(\mathcal{D})$, which is the unit simplex. Our analysis relies heavily on this special structure of $(\mathcal{D})$. In particular, any linear function can be easily optimized over the unit simplex which is required at each iteration in our algorithm. Therefore, we adopt the optimization formulation presented in this section.

3. Initial Approximation

In this section, we describe a procedure to compute an initial feasible solution of $(\mathcal{D})$ whose objective function value provides a good approximation of the optimal value.

As observed in Megiddo (1983), the weighted Euclidean one-center problem has the following geometric interpretation: Given $\rho > 0$, consider the balls defined by

$$\mathcal{B}_i(\rho) = \left\{ x \in \mathbb{R}^n : \|x - a^i\| \le \frac{\rho}{\omega_i} \right\}, \quad i = 1, \ldots, m.$$

Let $(c_{\mathcal{A}}, \rho_{\mathcal{A}})$ denote the optimal solution of $(\mathcal{P}_1)$. Then, $\rho_{\mathcal{A}}$ is the smallest value of ρ such that the balls $\mathcal{B}_i(\rho)$ have a nonempty intersection and $c_{\mathcal{A}}$ is the unique point in the intersection of the balls $\mathcal{B}_i(\rho_{\mathcal{A}})$.

Motivated by this geometric interpretation, let $a^j \in \mathcal{A}$ be the point with the largest corresponding weight ω_j.

We now construct balls $\mathcal{B}_i(\rho)$ for increasing values of $\rho > 0$. For each $i = 1, \ldots, m$, $i \neq j$, there exists a unique value $\rho^i > 0$ such that the balls $\mathcal{B}_i(\rho)$ and $\mathcal{B}_j(\rho)$ intersect for the first time when $\rho = \rho^i$. Let $\rho_* := \max_{i=1,\ldots,m,\, i\neq j} \rho^i > 0$. It follows from the geometric interpretation above that $\rho_* \leq \rho_{\mathcal{A}}$. It turns out that ρ_* is a provably good approximation to $\rho_{\mathcal{A}}$.

We describe the procedure more formally in Algorithm 3.1.

Algorithm 3.1
The algorithm that computes an initial feasible solution of $(\mathcal{D})$:

```
Require: Input set of points $\mathcal{A} = \{a^1, \ldots, a^m\} \subset \mathbb{R}^n$,
    $\mathcal{W} = \{\omega_1, \ldots, \omega_m\}$.
$j \leftarrow \arg\max_{i=1,\ldots,m} \omega_i$;
for all $i$ such that $1 \leq i \leq m$, $i \neq j$ do
    $\rho^i \leftarrow \|a^i - a^j\|/(1/\omega_i + 1/\omega_j)$;
end for
$\rho_* \leftarrow \max_{i=1,\ldots,m,\, i\neq j} \rho^i$; $j_* \leftarrow \arg\max_{i=1,\ldots,m,\, i\neq j} \rho^i$;
$u^0 \leftarrow 0$; $u^0_j \leftarrow \omega_{j_*}/(\omega_{j_*} + \omega_j)$; $u^0_{j_*} \leftarrow \omega_j/(\omega_{j_*} + \omega_j)$;
Output: $u^0$, $a^j$, $a^{j_*}$.
```

Lemma 3.1. *Algorithm* 3.1 *computes a feasible solution* $u^0 \in \mathbb{R}^m$ *of* $(\mathcal{D})$ *in* $O(mn)$ *arithmetic operations such that*

$$\Phi(u^0) \leq \Phi(u^*) = \gamma_{\mathcal{A}} \leq 9\Phi(u^0). \tag{8}$$

Proof. Clearly, Algorithm 3.1 terminates in $O(mn)$ operations. Note that the first inequality in (8) simply follows from the fact that $u^0 \in \mathbb{R}^m$ is a feasible solution of the maximization problem $(\mathcal{D})$.

It is easy to verify that

$$\begin{aligned} &\|(1-\sigma)y + \sigma z\|^2 \\ &\quad = (1-\sigma)\|y\|^2 + \sigma\|z\|^2 - \sigma(1-\sigma)\|y-z\|^2 \end{aligned} \tag{9}$$

for all $y, z \in \mathbb{R}^m$, and $\sigma \in \mathbb{R}$.

Let us define $\sigma = (u^0_{j_*}\nu_{j_*})/(u^0_j\nu_j + u^0_{j_*}\nu_{j_*})$. To prove the second inequality in (8), we have

$$\begin{aligned} \Phi(u^0) &= u^0_j\nu_j\|a^j\|^2 + u^0_{j_*}\nu_{j_*}\|a^{j_*}\|^2 \\ &\quad - \frac{\left(u^0_j\nu_j a^j + u^0_{j_*}\nu_{j_*}a^{j_*}\right)^T\left(u^0_j\nu_j a^j + u^0_{j_*}\nu_{j_*}a^{j_*}\right)}{u^0_j\nu_j + u^0_{j_*}\nu_{j_*}} \\ &= u^0_j\nu_j\|a^j\|^2 + u^0_{j_*}\nu_{j_*}\|a^{j_*}\|^2 \\ &\quad - \left(u^0_j\nu_j + u^0_{j_*}\nu_{j_*}\right)\|(1-\sigma)a^j + \sigma a^{j_*}\|^2 \\ &= u^0_j\nu_j\|a^j\|^2 + u^0_{j_*}\nu_{j_*}\|a^{j_*}\|^2 - u^0_j\nu_j\|a^j\|^2 - u^0_{j_*}\nu_{j_*}\|a^{j_*}\|^2 \\ &\quad + \frac{u^0_j\nu_j u^0_{j_*}\nu_{j_*}}{u^0_j\nu_j + u^0_{j_*}\nu_{j_*}}\|a^j - a^{j_*}\|^2 \\ &= \frac{u^0_j\nu_j u^0_{j_*}\nu_{j_*}}{u^0_j\nu_j + u^0_{j_*}\nu_{j_*}}\|a^j - a^{j_*}\|^2 \\ &= \frac{\|a^j - a^{j_*}\|^2}{(1/\omega_j + 1/\omega_{j_*})^2} \\ &= \rho_*^2, \end{aligned}$$

where we used (9) in the third line and (4) in the next-to-last one.

For each $i = 1, \ldots, m, i \neq j$, ρ^i is the optimal value of the weighted Euclidean one-center problem for the instance $(\{a^i, a^j\}, \{\omega_i, \omega_j\})$. Let $c^0 \in \mathbb{R}^n$ denote the optimal weighted center of the instance $(\{a^{j_*}, a^j\}, \{\omega_{j_*}, \omega_j\})$. It is easy to verify that $c^0 = \alpha a^{j_*} + (1-\alpha)a^j$, where $\alpha := \omega_{j_*}/(\omega_j + \omega_{j_*})$. For any $i = 1, \ldots, m$, we have

$$\begin{aligned} \|c^0 - a^i\| &\leq \|c^0 - a^j\| + \|a^j - a^i\| \\ &= \frac{\rho_*}{\omega_j} + \rho^i\left(\frac{1}{\omega_j} + \frac{1}{\omega_i}\right) \\ &\leq \rho_*\left(\frac{2}{\omega_j} + \frac{1}{\omega_i}\right) \\ &\leq \rho_*\left(\frac{3}{\omega_i}\right), \end{aligned}$$

where we used the inequalities $\rho^i \leq \rho_*$ and $\omega_i \leq \omega_j$ in the third line and the last line, respectively. It follows then that

$$\omega_i\|c^0 - a^i\| \leq 3\rho_*, \quad i = 1, \ldots, m.$$

This implies that $(c, \rho) := (c^0, 3\rho_*)$ is a feasible solution of $(\mathcal{P}_1)$ and the second inequality in (8) immediately follows. □

It follows from Lemma 3.1 that Algorithm 3.1 is a simple 3-approximation algorithm for the weighted Euclidean one-center problem. Drezner and Gavish (1985, Theorem 1) propose a very similar algorithm and establish that $(a^j, \rho(a^j))$ is a 2-approximate solution, where $\rho(\cdot)$ is defined as in (2) and j is the index of the point in $\mathcal{A}$ with the maximum weight. In the context of the dual problem $(\mathcal{D})$, the feasible solution produced by their algorithm is given by $u^0 = e^j$. Since $\Phi(u^0) = 0$, the objective function value of this initial feasible solution cannot be used to obtain an upper bound on the optimal value $\Phi(u^*)$ of $(\mathcal{D})$ such as that given by Lemma 3.1.

4. Existence of a Core Set

In this section, we establish the existence of a core set of size $\mathcal{O}(1/(\tau\epsilon))$ for the weighted Euclidean one-center problem, where τ is the squared ratio of the smallest weight to the largest weight. Our analysis mimics and extends the analysis of Bădoiu and Clarkson (2003), which demonstrates the existence of a core set of size $\mathcal{O}(1/\epsilon)$ for the minimum enclosing ball problem. The main ingredient in their analysis is the so-called "halfspace lemma," which states that every closed halfspace passing through the optimal center should contain at least one point on the boundary of the minimum enclosing ball. We start by extending this result to the weighted case. In contrast to the previous proofs of the halfspace lemma, we establish the following more general result as an immediate consequence of the optimality conditions (5).

Lemma 4.1. *Let $(c_{\mathcal{A}}, \rho_{\mathcal{A}})$ denote the optimal solution of a given instance $(\mathcal{A}, \mathcal{W})$ of the weighted Euclidean one-center problem. Every closed halfspace passing through $c_{\mathcal{A}}$ contains at least one point $a^i \in \mathcal{A}$ such that $\omega_i \|a^i - c_{\mathcal{A}}\| = \rho_{\mathcal{A}}$.*

Proof. By (5b), $c_{\mathcal{A}}$ lies in the convex hull of a subset of the input points given by $\mathcal{X} := \{a^j \in \mathcal{A}\colon u_j^* > 0\}$, where $u^* \in \mathbb{R}^m$ denotes any optimal solution of $(\mathcal{D})$. Hence, every closed halfspace passing through $c_{\mathcal{A}}$ must contain at least one point $a^i \in \mathcal{X}$. By (5d), each point $a^i \in \mathcal{X}$ satisfies $\omega_i \|a^i - c_{\mathcal{A}}\| = \rho_{\mathcal{A}}$, which completes the proof. □

We are now ready to prove the existence of a core set for the weighted Euclidean one-center problem.

Lemma 4.2. *Given an instance $(\mathcal{A}, \mathcal{W})$ of the weighted Euclidean one-center problem and $\epsilon \in (0, 1)$, there exists an ϵ-core set $\mathcal{X} \subseteq \mathcal{A}$ of size $\mathcal{O}(1/(\tau\epsilon^2))$, where $\tau \in (0, 1)$ is the squared ratio of the smallest weight to the largest weight.*

Proof. We proceed in a similar manner as in Bădoiu and Clarkson (2003). Initially, we set $\mathcal{X}_0 := \{a^j, a^{j_*}\} \subset \mathcal{A}$, where a^j and a^{j_*} are the two points computed by Algorithm 3.1. At iteration k, let (c^k, ρ^k) denote the optimal solution for the reduced instance $(\mathcal{X}_k, \mathcal{W}_k)$ of the weighted Euclidean one-center problem, where $\mathcal{W}_k := \{\omega_i\colon a^i \in \mathcal{X}_k\}$. If $\omega_j \|a^j - c^k\| \leq (1+\epsilon)\rho^k$ for each $a^j \in \mathcal{A}$, then $\mathcal{X}_k$ is an ϵ-core set since $\rho^k \leq \rho_{\mathcal{A}} \leq (1+\epsilon)\rho^k$. Otherwise, let $a^{k^*} \in \mathcal{A}$ denote the point with the largest weighted Euclidean distance from c^k. Then, we set $\mathcal{X}_{k+1} = \mathcal{X}_k \cup \{a^{k^*}\}$, $\mathcal{W}_{k+1} = \mathcal{W}_k \cup \{\omega_{k^*}\}$, and continue in a similar manner using the optimal solution (c^{k+1}, ρ^{k+1}) of the new instance $(\mathcal{X}_{k+1}, \mathcal{W}_{k+1})$.

By Lemma 3.1, $\rho^0 \geq \rho_{\mathcal{A}}/3$. The proof is based on establishing that the sequence $\{\rho^k\}$ is strictly increasing and that the ratio ρ^{k+1}/ρ^k can be bounded away from one. Suppose that the termination criterion is not satisfied at iteration k. Let $\xi^k := \|c^{k+1} - c^k\|$, $k = 0, 1, \ldots,$ and let $\omega_{\max} := \max_{i=1,\ldots,m} \omega_i$. There are two cases:

Case 1. Suppose that

$$\xi^k < \frac{\epsilon\rho^k}{2\omega_{\max}}.$$

In this case, we have, by the triangle inequality,

$$\|a^{k^*} - c^k\| \leq \|a^{k^*} - c^{k+1}\| + \xi^k < \|a^{k^*} - c^{k+1}\| + \frac{\epsilon\rho^k}{2\omega_{\max}},$$

which implies that

$$\begin{aligned}\frac{\rho^{k+1}}{\omega_{k^*}} &\geq \|a^{k^*} - c^{k+1}\| > \|a^{k^*} - c^k\| - \frac{\epsilon\rho^k}{2\omega_{\max}} \\ &> \frac{(1+\epsilon)\rho^k}{\omega_{k^*}} - \frac{\epsilon\rho^k}{2\omega_{\max}} \geq \frac{(1+\epsilon/2)\rho^k}{\omega_{k^*}},\end{aligned}$$

where we used $a^{k^*} \in \mathcal{X}_{k+1}$ to derive the first inequality, the fact that the termination criterion is not satisfied at iteration k to obtain the third inequality, and $\omega_{k^*} \leq \omega_{\max}$ to arrive at the last one. It follows that

$$\rho^{k+1} > \left(1 + \frac{\epsilon}{2}\right)\rho^k \geq \left(1 + \frac{\tau\epsilon^2}{9}\right)\rho^k, \qquad (10)$$

where we used the facts that $\epsilon \in (0, 1)$ and $\tau \in (0, 1]$.

Case 2. Suppose now that

$$\xi^k \geq \frac{\epsilon\rho^k}{2\omega_{\max}}.$$

Let $\mathcal{H}$ denote the hyperplane passing through c^k and perpendicular to $c^{k+1} - c^k$, and let $\mathcal{H}_-$ denote the closed halfspace bounded by $\mathcal{H}$ and not containing c^{k+1}. By Lemma 4.1, $\mathcal{H}_-$ contains a point $a^i \in \mathcal{X}_k$ such that $\omega_i \|a^i - c^k\| = \rho^k$. Therefore, for this input point,

$$\|a^i - c^{k+1}\|^2 \geq \|a^i - c^k\|^2 + (\xi^k)^2 \geq \left(\frac{\rho^k}{\omega_i}\right)^2 + \left(\frac{\epsilon\rho^k}{2\omega_{\max}}\right)^2.$$

Since $a^i \in \mathcal{X}_{k+1}$, it follows that

$$\frac{\rho^{k+1}}{\omega_i} \geq \|a^i - c^{k+1}\| \geq \frac{\rho^k}{\omega_i}\sqrt{1 + \left(\frac{\epsilon\omega_i}{2\omega_{\max}}\right)^2}.$$

Hence, we obtain

$$\rho^{k+1} \geq \rho^k\sqrt{1 + \frac{\tau\epsilon^2}{4}} \geq \left(1 + \frac{\tau\epsilon^2}{9}\right)\rho^k, \qquad (11)$$

where we used the definition of τ and the facts that $\epsilon \in (0, 1)$ and $\tau \in (0, 1]$.

By (10), (11), and Lemma 3.1, we have

$$\rho_{\mathcal{A}} \geq \rho^k \geq \left(1 + \frac{\tau\epsilon^2}{9}\right)^k \rho^0 \geq \frac{1}{3}\left(1 + \frac{\tau\epsilon^2}{9}\right)^k \rho_{\mathcal{A}}, \qquad (12)$$

which implies that the total number of iterations in this procedure is bounded above by

$$\frac{\log 3}{\log(1 + \tau\epsilon^2/9)} \leq (\log 3)\left(1 + \frac{9}{\tau\epsilon^2}\right),$$

where we used the inequality $\log(1+x) \geq x/(x+1)$ for $x > -1$. The assertion follows from the facts that $|\mathcal{X}_0| = 2$ and that each iteration adds one point to the working core set. □

Using a more careful bookkeeping argument as in Kumar et al. (2003), we establish that an improved bound on the core set size can be obtained with the same procedure.

Theorem 4.1. *Given an instance $(\mathscr{A}, \mathscr{W})$ of the weighted Euclidean one-center problem and $\epsilon \in (0,1)$, there exists an ϵ-core set $\mathscr{X} \subseteq \mathscr{A}$ of size $\mathscr{O}(1/(\tau\epsilon))$.*

Proof. For the procedure outlined in the proof of Lemma 4.2, let us define

$$\psi(i) := \min\{k\colon \mathscr{X}_k \text{ is a } (1/2^i)\text{-core set}\}, \quad i = 1, 2, \ldots.$$

By Lemma 4.2, $\psi(1) = \mathscr{O}(1/\tau)$. For $i \geq 2$, we derive an upper bound on $\psi(i) - \psi(i-1)$. Note that $\mathscr{X}_{\psi(i-1)}$ is a $(1/2^{(i-1)})$-core set. It follows from (12) that

$$\begin{aligned}\rho_{\mathscr{A}} \geq \rho^{\psi(i)} &\geq \rho^{\psi(i-1)}\left(1 + \frac{\tau}{9(2^{2i})}\right)^{(\psi(i)-\psi(i-1))}\\ &\geq \frac{\rho_{\mathscr{A}}}{1+1/2^{(i-1)}}\left(1 + \frac{\tau}{9(2^{2i})}\right)^{(\psi(i)-\psi(i-1))},\end{aligned}$$

which implies that

$$\begin{aligned}\psi(i) - \psi(i-1) &\leq \frac{\log(1+1/2^{i-1})}{\log(1+\tau/(9(2^{2i+4})))}\\ &\leq \left(\frac{1}{2^{i-1}}\right)\left(1 + \frac{9(2^{2i})}{\tau}\right) = \mathscr{O}(2^i/\tau),\end{aligned}$$

where we used the inequalities $\log(1+x) \leq x$ and $1/\log(1+x) \leq 1 + 1/x$. Note that $\mathscr{X}_k$ is an ϵ-core set after $\Psi(\lceil \log_2(1/\epsilon)\rceil)$ iterations. Therefore, the total number of iterations can be bounded above by

$$\begin{aligned}\Psi(\lceil \log_2(1/\epsilon)\rceil) &= \Psi(1) + \sum_{i=2}^{\lceil \log_2(1/\epsilon)\rceil} (\Psi(i) - \Psi(i-1))\\ &= \mathscr{O}(1/\tau) + \sum_{i=1}^{\lceil \log_2(1/\epsilon)\rceil} \mathscr{O}(2^i/\tau)\\ &= \mathscr{O}\left(\frac{2^{\lceil \log_2(1/\epsilon)\rceil}}{\tau}\right) = \mathscr{O}(1/(\tau\epsilon)).\end{aligned}$$

Arguing similarly as in the proof of Lemma 4.2, we obtain an ϵ-core set of size $\mathscr{O}(1/(\tau\epsilon))$. □

We remark that the procedure that yields the improved core set result of Theorem 4.1 can be turned into an efficient approximation algorithm under the assumption that the smaller instances of the weighted Euclidean one-center problem can be solved exactly and efficiently. In the next section, we propose and analyze an approximation algorithm that computes a core set satisfying the same asymptotic bound of Theorem 4.1 without the strong requirements of an exact and efficient solver for smaller subproblems.

In addition to establishing the existence of a core set of size $\mathscr{O}(1/\epsilon)$, Bădoiu and Clarkson (2003) also propose the following simple, iterative algorithm for the minimum enclosing ball problem. Their algorithm starts with any input point as the initial center c^1 and updates the center using the formula $c^{k+1} \leftarrow (1 - 1/(k+1))c^k + (1/(k+1))a^{k^*}$, where a^{k^*} denotes the furthest point from c^k. We do not pursue the generalization of their algorithm to the weighted Euclidean one-center problem in this paper for the following reasons. First, they establish that this algorithm computes a $(1+\epsilon)$-approximate solution to the minimum enclosing ball problem in $1/\epsilon^2$ iterations, which results in an overall complexity of $\mathscr{O}(mn/\epsilon^2)$ operations (Bădoiu and Clarkson 2003). In contrast, the specialization of our algorithm to the minimum enclosing ball problem requires only $\mathscr{O}(mn/\epsilon)$ operations. Second, their algorithm exclusively works with the primal problem $(\mathscr{P}_1)$. On the other hand, while our algorithm primarily works with the dual problem $(\mathscr{D})$, the termination criterion relies on the primal perspective. As such, the termination criterion may be satisfied earlier in our algorithm, whereas their algorithm requires exactly $1/\epsilon^2$ iterations in order to guarantee a $(1+\epsilon)$-approximate solution. Finally, the analysis of their algorithm relies on the following crucial property. At iteration k of their algorithm, suppose that $c^k \neq c^*$, where c^* is the optimal center. Let $\mathscr{H}$ denote the halfspace passing through c^* and perpendicular to $c^k - c^*$. Using the halfspace lemma, Bădoiu and Clarkson (2003) show that the furthest point a^{k^*} lies in the closed halfspace bounded by $\mathscr{H}$ and not containing c^k. Based on this observation, they can bound $\|c^{k+1} - c^*\|$ using the bound on $\|c^k - c^*\|$. The straightforward extension of this result to the weighted Euclidean one-center problem would require that the input point with the largest weighted distance from c^k would similarly lie in the closed halfspace bounded by $\mathscr{H}$ and not containing c^k. Despite the fact that the halfspace lemma can be extended to the weighted case (see Lemma 4.1), it turns out that this straightforward extension does not hold true as illustrated by the following simple instance. Let

$$\begin{aligned}\mathscr{A} &:= \{(1,0)^T, (-1,0)^T, (0.1253, 0.2877)^T\},\\ \mathscr{W} &:= \{1, 1, 3.1868\}.\end{aligned}$$

It is easy to verify that $c_{\mathscr{A}} = (0,0)^T$ and $\rho_{\mathscr{A}} = 1$. Suppose that $c^1 = (1,0)^T$. The weighted distance between c^1 and $(0.1253, 0.2877)^T$ is about 2.9344 while the weighted distance between c^1 and $(-1,0)^T$ is 2. Therefore, the input point with the largest weighted distance from c^1 does not have a nonpositive x_1 component, which reveals that the extension of the aforementioned result does not hold in general. This simple example illustrates that the main ingredient used in the analysis of their algorithm does not necessarily extend to the weighted Euclidean one-center problem. Therefore, even if the algorithm can be extended, the analysis would require a different approach, in which case, this would no longer be a straightforward extension of their result.

5. The Algorithm

In this section, given an input set $\mathcal{A} := \{a^1, \ldots, a^m\} \subset \mathbb{R}^n$ with corresponding positive weights $\mathcal{W} = \{\omega_1, \ldots, \omega_m\}$ and $\epsilon > 0$, we present an algorithm that computes a $(1+\epsilon)$-approximate solution to the weighted Euclidean one-center problem by approximately solving the dual problem ($\mathcal{D}$) (see Algorithm 5.1).

Algorithm 5.1

The algorithm that computes a $(1+\epsilon)$-approximate solution to the weighted Euclidean one-center of $(\mathcal{A}, \mathcal{W})$:

```
**Require:** Input set of points $\mathcal{A} = \{a^1, \ldots, a^m\} \subset \mathbb{R}^n$,
  $\mathcal{W} = \{\omega_1, \ldots, \omega_m\}$, $\epsilon > 0$.
Run Algorithm 3.1 to compute $u^0 \in \mathbb{R}^m$, $a^j$, $a^{j_*}$.
$\mathcal{X}_0 \leftarrow \{a^j, a^{j_*}\}$; $\nu_i \leftarrow (\omega_i)^2$, $i = 1, \ldots, m$;
$c^0 \leftarrow (1/\sum_{i=1}^m u_i^0 \nu_i) \sum_{i=1}^m u_i^0 \nu_i a^i$; $k \leftarrow 0$;
**repeat**
    $\gamma^k \leftarrow \Phi(u^k)$;
    $k^* \leftarrow \arg\max_{i=1,\ldots,m} \nu_i \|a^i - c^k\|^2$;
    $k_* \leftarrow \arg\min_{i:\, u_i^k > 0} \nu_i \|a^i - c^k\|^2$;
    $\delta_k^+ \leftarrow (\nu_{k^*} \|a^{k^*} - c^k\|^2/\gamma^k) - 1$,
    $\delta_k^- \leftarrow 1 - (\nu_{k_*} \|a^{k_*} - c^k\|^2/\gamma^k)$;
    $\delta_k \leftarrow \max\{\delta_k^+, \delta_k^-\}$;
    **if** $\delta_k \le (1+\epsilon)^2 - 1$ **then break**
    **if** $\delta_k > \delta_k^-$ **then**
        $\mu^k \leftarrow (\sum_{i=1}^m u_i^k \nu_i)/\nu_{k^*}$;
        $\lambda^k \leftarrow \begin{cases} \dfrac{\mu^k}{1-\mu^k}\left(\sqrt{1+\dfrac{(1-\mu^k)\delta_k}{1+\mu^k\delta_k}}-1\right) & \text{if } \mu^k < 1, \\ \delta_k/[2(1+\delta_k)] & \text{if } \mu^k = 1, \\ \dfrac{\mu^k}{\mu^k-1}\left(1-\sqrt{1-\dfrac{(\mu^k-1)\delta_k}{1+\mu^k\delta_k}}\right) & \text{if } \mu^k > 1; \end{cases}$
        $u^{k+1} \leftarrow (1-\lambda^k)u^k + \lambda^k e^{k^*}$;
        $c^{k+1} \leftarrow [1/((1-\lambda^k)\mu^k + \lambda^k)]$
                $\cdot ((1-\lambda^k)\mu^k c^k + \lambda^k a^{k^*})$;
        $\mathcal{X}_{k+1} \leftarrow \mathcal{X}_k \cup \{a^{k^*}\}$;
    **else**
        $\mu^k \leftarrow (\sum_{i=1}^m u_i^k \nu_i)/\nu_{k_*}$;
        $\lambda^k \leftarrow \begin{cases} +\infty \quad \text{if } \delta_k = 1, \\ \dfrac{\mu^k}{1-\mu^k}\left(1-\sqrt{1-\dfrac{(1-\mu^k)\delta_k}{1-\mu^k\delta_k}}\right) \\ \quad \text{if } \mu^k < 1, \\ \delta_k/[2(1-\delta_k)] \quad \text{if } \mu^k = 1, \\ \dfrac{\mu^k}{\mu^k-1}\left(\sqrt{1+\dfrac{(\mu^k-1)\delta_k}{1-\mu^k\delta_k}}-1\right) \\ \quad \text{if } \mu^k > 1 \text{ and } \mu^k\delta_k < 1, \\ +\infty \quad \text{if } \mu^k > 1 \text{ and } \mu^k\delta_k \ge 1; \end{cases}$
        $\lambda^k \leftarrow \min\{\lambda^k, u_{k_*}^k/(1-u_{k_*}^k)\}$;
        **if** $\lambda^k = u_{k_*}^k/(1-u_{k_*}^k)$ **then**
            $\mathcal{X}_{k+1} \leftarrow \mathcal{X}_k \backslash \{a^{k_*}\}$;
        **else**
            $\mathcal{X}_{k+1} \leftarrow \mathcal{X}_k$;
        **end if**
        $u^{k+1} \leftarrow (1+\lambda^k)u^k - \lambda^k e^{k_*}$;
        $c^{k+1} \leftarrow [1/((1+\lambda^k)\mu^k - \lambda^k)]$
                $\cdot ((1+\lambda^k)\mu^k c^k - \lambda^k a^{k_*})$;
    **end if**
    $k \leftarrow k+1$;
**until** $\delta_{k-1} \le (1+\epsilon)^2 - 1$
**Output** $c^k$, $\mathcal{X}_k$, $u^k$, $[(1+\delta_k)\gamma^k]^{1/2}$.
```

We now explain Algorithm 5.1 in more detail. The algorithm is initialized by calling Algorithm 3.1 that computes an initial feasible solution $u^0 \in \mathbb{R}^m$ of the dual formulation ($\mathcal{D}$). At each iteration, Algorithm 5.1 maintains a dual feasible solution $u^k \in \mathbb{R}^m$ and computes a trial solution $(c^k, (\gamma^k)^{1/2}) = (c^k, (\Phi(u^k))^{1/2})$. By (7), this solution coincides with the optimal solution $(c_{\mathcal{A}}, \rho_{\mathcal{A}})$ if and only if u^k is an optimal solution of ($\mathcal{D}$). Otherwise, by dual feasibility of u^k, we have $(\gamma^k)^{1/2} < \rho_{\mathcal{A}}$.

At each iteration, Algorithm 5.1 computes two parameters δ_k^+ and δ_k^-. Note that δ_k^+ is the smallest value of δ such that $(c, \gamma^{1/2}) = (c^k, [(1+\delta)\gamma^k]^{1/2})$ is a feasible solution of the primal formulation ($\mathcal{P}_1$). Similarly, δ_k^- is the smallest value of δ such that $\omega_i\|a^i - c^k\| \ge [(1-\delta)\gamma^k]^{1/2}$ for all $a^i \in \mathcal{X}_k$. Since $\delta_k := \max\{\delta_k^+, \delta_k^-\} \ge \delta_k^+$, it follows that

$$(\gamma^k)^{1/2} \le \rho_{\mathcal{A}} \le (1+\delta_k)^{1/2}(\gamma^k)^{1/2}. \tag{13}$$

Following Todd and Yıldırım (2007), iteration k is called a plus-iteration if $\delta_k > \delta_k^-$. It is called a minus-iteration if $\delta_k \le \delta_k^-$ and $\lambda^k > u_{k_*}/(1-u_{k_*})$. Otherwise, we call it a drop-iteration since $\mathcal{X}_{k+1}$ is then obtained by removing a^{k_*} from $\mathcal{X}_k$.

At a plus-iteration, the next feasible solution $u^{k+1} \in \mathbb{R}^m$ is given by an appropriate convex combination of u^k and e^{k^*}. The weights used in the convex combination are determined by

$$\lambda^k = \arg\max_{\lambda \in [0,1]} \Phi((1-\lambda)u^k + \lambda e^{k^*}). \tag{14}$$

Note that $u^{k+1} = (1-\lambda^k)u^k + \lambda^k e^{k^*}$ is a feasible solution of ($\mathcal{D}$) and the algorithm computes the new trial solution $(c^{k+1}, (\gamma^{k+1})^{1/2})$ as a function of u^{k+1}. It turns out that c^{k+1} is obtained by moving c^k towards $a^{k^*} \in \mathcal{A}$ in this case.

At a minus- or drop-iteration, the next feasible solution u^{k+1} is obtained by moving u^k away from e^{k_*}. In this case, λ^k is given by

$$\lambda^k = \arg\max_{\lambda \in [0,\, u_{k_*}^k/(1-u_{k_*}^k)]} \Phi((1+\lambda)u^k - \lambda e^{k_*}). \tag{15}$$

Note that the range of λ is chosen to ensure the nonnegativity of u^{k+1}. In contrast with a plus-iteration, c^{k+1} is obtained by moving c^k *away* from $a^{k_*} \in \mathcal{A}$ at a minus- or drop-iteration.

Algorithm 5.1 is the adaptation of the Frank-Wolfe algorithm (Frank and Wolfe 1956) using Wolfe's away steps (Wolfe 1970) to the weighted Euclidean one-center problem using the initialization procedure given by Algorithm 3.1. This algorithm is a sequential linear programming algorithm for the dual problem ($\mathcal{D}$) and generates a sequence of feasible solutions with nondecreasing objective function values. At each iteration, the nonlinear objective function $\Phi(u)$ is linearized at the current feasible solution u^k. At a plus-iteration, the new feasible solution u^{k+1} is obtained by moving towards the vertex of the unit simplex that maximizes this linear approximation. At a minus- or drop-iteration, u^{k+1} is obtained by moving away from the vertex that minimizes the linear approximation, where the minimization is restricted to the smallest face of the unit simplex that contains u^k. In either case, the parameter λ^k is chosen so as to ensure the maximum improvement in the original objective function $\Phi(u)$.

We remark that Algorithm 5.1 reduces to Algorithm 4.1 of Yıldırım (2008) if all weights ω_i are identical. Furthermore, μ^k is always equal to one in this case, which implies that the optimal solution λ^k of each of the line search problems (14) and (15) has a much simpler expression. In the presence of nonidentical weights, it turns out that the expression for λ^k depends on the value of μ^k at each iteration.

5.1. Analysis of the Algorithm

We analyze Algorithm 5.1 in this section. Note that the objective function values $\Phi(u^k)$ of the iterates generated by Algorithm 5.1 are monotonically nondecreasing due to the choice of λ^k given by (14) at a plus-iteration and by (15) at a minus- or drop-iteration. First, we establish lower bounds on the improvement at each plus- or minus-iteration.

Lemma 5.1. *At each plus- or minus-iteration, we have*

$$\frac{\gamma^{k+1}}{\gamma^k} \geq \begin{cases} 1+\dfrac{\mu^k(\delta_k)^2}{4(1+\delta_k)}, & \text{if } \mu^k < 1, \\[2ex] 1+\dfrac{(\delta_k)^2}{4(1+\delta_k)}, & \text{otherwise.} \end{cases} \tag{16}$$

Proof. By definition of c^k,

$$\gamma^k = \Phi(u^k) = \sum_{i=1}^m u_i^k \nu_i \|a^i\|^2 - \sum_{i=1}^m u_i^k \nu_i \|c^k\|^2. \tag{17}$$

Let us first consider a plus-iteration. In this case, $u^{k+1} = (1-\lambda^k)u^k + \lambda^k e^{k^*}$, where $a^{k^*} \in \mathcal{A}$ is the point with the largest weighted distance from c^k. Furthermore, $c^{k+1} = (1-\sigma)c^k + \sigma a^{k^*}$, where $\sigma = \lambda^k/((1-\lambda^k)\mu^k + \lambda^k)$. Therefore,

$$\begin{aligned}
\gamma^{k+1} &= \Phi((1-\lambda^k)u^k + \lambda^k e^{k^*}) \\
&= (1-\lambda^k)\sum_{i=1}^m u_i^k \nu_i \|a^i\|^2 + \lambda^k \nu_{k^*}\|a^{k^*}\|^2 \\
&\quad -\left[(1-\lambda^k)\sum_{i=1}^m u_i^k\nu_i + \lambda^k\nu_{k^*}\right]\Big[(1-\sigma)\|c^k\|^2 \\
&\qquad\qquad + \sigma\|a^{k^*}\|^2 - \sigma(1-\sigma)\|a^{k^*}-c^k\|^2\Big] \\
&= (1-\lambda^k)\sum_{i=1}^m u_i^k\nu_i\|a^i\|^2 - \frac{(1-\sigma)\nu_{k^*}\lambda^k}{\sigma}\|c^k\|^2 \\
&\quad + \nu_{k^*}\lambda^k(1-\sigma)\|a^{k^*}-c^k\|^2 \\
&= (1-\lambda^k)\left[\sum_{i=1}^m u_i^k\nu_i\|a^i\|^2 - \sum_{i=1}^m u_i^k\nu_i\|c^k\|^2\right] \\
&\quad + \lambda^k(1-\sigma)(1+\delta_k)\gamma^k \\
&= \gamma^k(1-\lambda^k)\left[1+\frac{\lambda^k\mu^k(1+\delta_k)}{(1-\lambda^k)\mu^k+\lambda^k}\right],
\end{aligned}$$

where we used (9) for the computation of $\|c^{k+1}\|^2$ in the second equality, the definitions of μ^k and σ in the third one, and the definitions of μ^k and δ_k in the fourth one. It follows that

$$\gamma^{k+1} = \gamma^k\Delta_k^+(\lambda^k),$$

where

$$\Delta_k^+(\lambda) := (1-\lambda)\left(1+\frac{\lambda(1+\delta_k)}{1-\lambda+\lambda/\mu^k}\right).$$

It is straightforward to verify that the first and second derivatives of Δ_k^+ with respect to λ are given by

$$\begin{aligned}
&(\Delta_k^+)'(\lambda) \\
&\quad = (\lambda^2(1+\mu^k\delta_k)(\mu^k-1) - 2\lambda\mu^k(1+\mu^k\delta_k) + (\mu^k)^2\delta_k) \\
&\qquad /(\lambda(\mu^k-1)-\mu^k)^2, \\
&\qquad (\Delta_k^+)''(\lambda) = (2(1+\delta_k)(\mu^k)^2)/(\lambda(\mu^k-1)-\mu^k)^3,
\end{aligned}$$

which together imply that $\Delta_k^+(\lambda)$ is a strictly concave function on $\lambda \in [0,1]$ for each $\mu^k > 0$ and that $\lambda^k \in (0,1)$ is its unique maximizer.

The proof is based on establishing a lower bound on $\Delta_k^+(\lambda^k)$. Suppose first that $\mu^k < 1$. In this case, we have

$$\begin{aligned}
\lambda^k &= \frac{\mu^k}{1-\mu^k}\left(\sqrt{1+\frac{(1-\mu^k)\delta_k}{1+\mu^k\delta_k}}-1\right) \\
&= \frac{\mu^k\delta_k}{2\sqrt{1+\vartheta_1}(1+\mu^k\delta_k)} \\
&\geq \lambda_*^k := \frac{\mu^k\delta_k}{2(1+\delta_k)},
\end{aligned}$$

where we used the mean value theorem on the function $\sqrt{1+x}$ to derive the second equality with $\vartheta_1 \in (0, [(1-\mu^k)\delta_k]/(1+\mu^k\delta_k))$, and we used the upper bound on ϑ_1 and the fact that $\mu^k < 1$ to arrive at the last inequality.

Since λ^k is the maximizer of $\Delta_k^+(\lambda)$, it follows that

$$\begin{aligned}
&\Delta_k^+(\lambda^k)\\
&\quad\ge \Delta_k^+(\lambda_*^k) = 1+\frac{\mu^k(\delta_k)^2}{2(1+\delta_k)}\left(1-\frac{1+\delta_k}{2(1+\delta_k)+(1-\mu^k)\delta_k}\right)\\
&\quad\ge 1+\frac{\mu^k(\delta_k)^2}{4(1+\delta_k)},
\end{aligned}$$

where we used $\mu^k < 1$ to derive the last inequality. This establishes the first part of (16) at a plus-iteration.

Suppose now that $\mu^k = 1$. Since $\lambda^k = \delta_k/[2(1+\delta_k)]$, we have

$$\Delta_k^+(\lambda^k) = 1+\frac{(\delta_k)^2}{4(1+\delta_k)}.$$

Finally, if $\mu^k > 1$ at a plus-iteration, then we have

$$\begin{aligned}
\lambda^k &= \frac{\mu^k}{\mu^k-1}\left(1-\sqrt{1-\frac{(\mu^k-1)\delta_k}{1+\mu^k\delta_k}}\right)\\
&\ge \frac{\delta_k}{2((1/\mu^k)+\delta_k)} \ge \frac{\delta_k}{2(1+\delta_k)},
\end{aligned}$$

where we used the inequality $\sqrt{1-x} \le 1-(1/2)x$ for $x \le 1$ and the fact that $\mu^k > 1$. The second part of the inequality (16) follows from the previous case since $1-\lambda^k+(\lambda^k/\mu^k) < 1$, which completes the proof for a plus-iteration.

Let us now consider a minus-iteration. In this case, $u^{k+1} = (1+\lambda^k)u^k - \lambda^k e^{k_*}$, where $a^{k_*} \in \mathcal{X}_k$ is the point with the smallest weighted distance from c^k. Similarly to a plus-iteration, we obtain

$$\gamma^{k+1} = \Phi((1+\lambda^k)u^k - \lambda^k e^{k_*}) = \gamma^k \Delta_k^-(\lambda^k),$$

where

$$\Delta_k^-(\lambda) := (1+\lambda)\left(1-\frac{\lambda(1-\delta_k)}{1+\lambda-\lambda/\mu^k}\right).$$

Note that $\delta_k \in (0,1]$ at a minus-iteration. The first and second derivatives of Δ_k^- are given by

$$\begin{aligned}
&(\Delta_k^-)'(\lambda)\\
&\quad= (\lambda^2(\mu^k-1)(\mu^k\delta_k-1)+2\lambda\mu^k(\mu^k\delta_k-1)+(\mu^k)^2\delta_k)\\
&\qquad /(\mu^k+(\mu^k-1)\lambda)^2,\\
&\qquad (\Delta_k^-)''(\lambda) = (2(\delta_k-1)(\mu^k)^2)/(\mu^k+(\mu^k-1)\lambda)^3.
\end{aligned}$$

If $\delta_k = 1$, then $\Delta_k^-(\lambda) \to +\infty$ as $\lambda \to +\infty$. Similarly, if $\delta_k < 1$ and $\mu^k\delta_k \ge 1$, then $\Delta_k^-(\lambda)$ is a strictly increasing function on $\lambda \ge 0$. Therefore, Algorithm 5.1 sets $\lambda^k = +\infty$ in either one of these two cases, which subsequently leads to a drop-iteration.

Suppose first that $\mu^k < 1$. In this case, $\Delta_k^-(\lambda)$ is a strictly concave function on $\lambda \in [0, \mu^k/(1-\mu^k))$ since $\delta_k \in (0,1)$ at a minus-iteration. The unique maximizer λ^k is given by

$$\lambda^k = \frac{\mu^k}{1-\mu^k}\left(1-\sqrt{1-\frac{(1-\mu^k)\delta_k}{1-\mu^k\delta_k}}\right) \ge \lambda_{**}^k := \frac{\mu^k\delta_k}{2(1-\mu^k\delta_k)},$$

where we again used the inequality $\sqrt{1-x} \le 1-(1/2)x$ for $x \le 1$. Therefore,

$$\Delta_k^-(\lambda^k) \ge \Delta_k^-(\lambda_{**}^k) = 1+\frac{\mu^k(\delta_k)^2}{2(2-\delta_k(\mu^k+1))} \ge 1+\frac{\mu^k(\delta_k)^2}{4(1+\delta_k)}.$$

This establishes the first part of (16) at a minus-iteration.

Suppose now that $\mu^k = 1$. Since $\lambda^k = \delta_k/[2(1-\delta_k)]$ at a minus-iteration, we have

$$\Delta_k^-(\lambda^k) = 1+\frac{(\delta_k)^2}{4(1-\delta_k)} \ge 1+\frac{(\delta_k)^2}{4(1+\delta_k)}.$$

Finally, if $\mu^k > 1$ at a minus-iteration, note that we should necessarily have $\mu^k\delta_k < 1$. In this case, $\Delta_k^-(\lambda)$ is a strictly concave function on $\lambda \ge 0$, and the unique maximizer λ^k is given by

$$\begin{aligned}
\lambda^k &= \frac{\mu^k}{\mu^k-1}\left(\sqrt{1+\frac{(\mu^k-1)\delta_k}{1-\mu^k\delta_k}}-1\right)\\
&= \frac{\mu^k\delta_k}{2\sqrt{1+\vartheta_2}(1-\mu^k\delta_k)}\\
&\ge \frac{\delta_k}{2(1-\delta_k)},
\end{aligned}$$

where we once again invoked the mean value theorem with $\vartheta_2 \in (0, [(\mu^k-1)\delta_k]/[1-\mu^k\delta_k])$ to derive the second equality, and we used the upper bound on ϑ_2 and the fact that $\mu^k > 1$ to obtain the inequality.

The second part of the inequality (16) follows from the previous case since $1+\lambda^k-(\lambda^k/\mu^k) > 1$, which completes the proof. □

Note that Lemma 5.1 establishes lower bounds on the improvement at each plus- or minus-iteration. On the other hand, no such lower bound can be derived for drop-iterations since λ^k can be arbitrarily small. Therefore, we can only say that the dual objective function value does not decrease at a drop-iteration.

We remark that the lower bounds on the improvement at each plus- or minus-iteration depend on μ^k. The following result is an immediate consequence of Lemma 5.1.

Corollary 5.1. *Let* $\mu^* := \min\{1, \inf_{k=0,1,\ldots}\mu^k\} > 0$. *Then, at each plus- or minus-iteration,*

$$\frac{\gamma^{k+1}}{\gamma^k} \ge 1+\frac{\mu^*(\delta_k)^2}{4(1+\delta_k)}. \tag{18}$$

We next analyze the complexity of Algorithm 5.1. For $\delta > 0$, let us define the following parameter:

$$\theta(\delta) := \min\{k\colon \delta_k \le \delta\}. \tag{19}$$

Also, we denote the number of drop-iterations in the first $\theta(\delta)$ iterations of Algorithm 5.1 by $\phi(\delta)$.

Lemma 5.2. *$\theta(\cdot)$ and $\phi(\cdot)$ satisfy the following relationships*:

$$\phi(1) = 0, \tag{20a}$$

$$\theta(1) = \mathcal{O}(1/\mu^*), \tag{20b}$$

$$\theta(1/2^i) - \theta(1/2^{i-1}) = \mathcal{O}(2^i/\mu^*) + (\phi(1/2^i) - \phi(1/2^{i-1})), \quad i = 1, 2, \ldots. \tag{20c}$$

Proof. Note that Algorithm 5.1 cannot have any minus- or drop-iterations until $\delta_k \le 1$, which implies that $\phi(1) = 0$. Therefore, at each plus-iteration k with $\delta_k > 1$, it follows from Corollary 5.1 that

$$\frac{\gamma^{k+1}}{\gamma^k} \ge 1 + \frac{\mu^*(\delta_k)^2}{4(1+\delta_k)} \ge 1 + \frac{\mu^*}{8},$$

where we used the fact that $x^2/(1+x)$ is an increasing function on $x \ge 0$. Iterating the inequality above and using the fact that $9\gamma^0 \ge \gamma_{\mathcal{A}} \ge \gamma^0$ (see Lemma 3.1), we obtain

$$\gamma_{\mathcal{A}} \ge \gamma^k \ge [1 + (\mu^*/8)]^k \gamma^0 \ge [1 + (\mu^*/8)]^k (\gamma_{\mathcal{A}}/9),$$

which implies that $\theta(1) = \mathcal{O}(\log 9/(\log(1 + (\mu^*/8)))) = \mathcal{O}(1/\mu^*)$, where we used the inequality $\log(1+x) \ge x/(x+1)$ for all $x > -1$. This establishes (20b).

Let i be any positive integer and let $\tilde{k} := \theta(1/2^{i-1})$. At each plus- or minus-iteration with $\delta_k > 1/2^i$, it follows from Corollary 5.1 that

$$\gamma^{k+1} \ge \gamma^k\left(1 + \frac{\mu^*(\delta_k)^2}{4(1+\delta_k)}\right) \ge \gamma^k\left(1 + \frac{\mu^*}{2^{i+2}(2^i+1)}\right).$$

At a drop-iteration, we only have $\gamma^{k+1} \ge \gamma^k$. Therefore, let $\phi_i := \phi(1/2^i) - \phi(1/2^{i-1})$ denote the number of drop-iterations between iteration number $\theta(1/2^{i-1})$ and iteration number $\theta(1/2^i)$ of Algorithm 5.1. Therefore, iterating the above inequality and using the fact that $[1 + (1/2^{i-1})]\gamma^{\tilde{k}} \ge \gamma_{\mathcal{A}} \ge \gamma^{\tilde{k}}$ (see (13)), we can bound the number of plus- or minus-iterations ζ between iteration $\theta(1/2^i)$ and iteration $\theta(1/2^{i-1})$ using

$$\begin{aligned}\gamma_{\mathcal{A}} \ge \gamma^{\tilde{k}+\zeta+\phi_i} &\ge \gamma^{\tilde{k}}\left(1 + \frac{\mu^*}{2^{i+2}(2^i+1)}\right)^{\zeta} \\ &\ge \frac{\gamma_{\mathcal{A}}}{1 + (1/2^{i-1})}\left(1 + \frac{\mu^*}{2^{2+i}(2^i+1)}\right)^{\zeta},\end{aligned}$$

which implies that

$$\begin{aligned}\zeta + \phi_i &\le \frac{\log(1 + 1/2^{i-1})}{\log(1 + \mu^*/(2^{2+i}(2^i+1)))} + \phi_i \\ &\le \left(\frac{1}{2^{i-1}}\right)\left(1 + \frac{2^{2+i}(2^i+1)}{\mu^*}\right) + \phi_i \\ &= \mathcal{O}(2^i/\mu^*) + \phi_i,\end{aligned}$$

where we used the inequalities $\log(1+x) \le x$ and $\log(1+x) \ge x/(x+1)$. This implies that $\theta(1/2^i) - \theta(1/2^{i-1}) = \mathcal{O}(2^i/\mu^*) + (\phi(1/2^i) - \phi(1/2^{i-1}))$, which completes the proof. □

We are now in a position to establish the iteration complexity of Algorithm 5.1.

Lemma 5.3. *Let $\epsilon \in (0,1)$. Then, Algorithm 5.1 computes a $(1+\epsilon)$-approximate solution in $\theta(\epsilon) = \mathcal{O}(1/(\mu^*\epsilon))$ iterations.*

Proof. Let i^* be a positive integer such that $1/2^{i^*} \le \epsilon < 1/2^{i^*-1}$. Therefore, $\theta(\epsilon) \le \theta(1/2^{i^*})$. By Proposition 5.2,

$$\begin{aligned}&\theta(1/2^{i^*}) \\ &\quad = \theta(1) + \sum_{i=1}^{i^*}(\theta(1/2^i) - \theta(1/2^{i-1})) \\ &\quad = \mathcal{O}(1/\mu^*) + \sum_{i=1}^{i^*}(\mathcal{O}(2^i/\mu^*) + (\phi(1/2^i) - \phi(1/2^{i-1}))) \\ &\quad = \mathcal{O}(1/(\mu^*\epsilon)) + \phi(1/2^{i^*}),\end{aligned}$$

where we used the fact that $2^{i^*} < 2/\epsilon$.

The proof will be complete if we can establish that $\phi(1/2^{i^*}) = \mathcal{O}(1/(\mu^*\epsilon))$. Note that we cannot bound the improvement from below at a drop-iteration. However, each such iteration can be coupled with the latest previous plus-iteration in which the component of u that just dropped to zero is increased from zero. To account for the two initial positive components of u^0, we may have to increase the iteration count by two. It follows that $\phi(1/2^{i^*}) = \mathcal{O}(1/(\mu^*\epsilon))$. □

The following theorem establishes the overall complexity of Algorithm 5.1.

Theorem 5.1. *Given $\mathcal{A} := \{a^1, \ldots, a^m\} \subset \mathbb{R}^n$ with corresponding weights $\mathcal{W} = \{\omega_1, \ldots, \omega_m\}$ and $\epsilon \in (0,1)$, Algorithm 5.1 computes a $(1+\epsilon)$-approximate solution for the instance $(\mathcal{A}, \mathcal{W})$ of the weighted Euclidean one-center problem in $\mathcal{O}(mn/(\mu^*\epsilon))$ arithmetic operations.*

Proof. Let u^η denote the final iterate computed by Algorithm 5.1 and let $\gamma^\eta = \Phi(u^\eta)$. By (13),

$$\gamma^\eta \le \gamma_{\mathcal{A}} \le (1 + \delta_\eta)\gamma^\eta.$$

Since $\delta_\eta \le (1+\epsilon)^2 - 1$ by the termination criterion, it follows that $(\gamma^\eta)^{1/2} \le \rho_{\mathcal{A}} \le [(1+\delta_\eta)\gamma^\eta]^{1/2} \le (1+\epsilon)(\gamma^\eta)^{1/2}$, which implies that $(c^\eta, \rho(c^\eta)) = (c^\eta, [(1+\delta_\eta)\gamma^\eta]^{1/2})$ is a $(1+\epsilon)$-approximate solution.

At each iteration, the dominating work is the computation of the largest weighted distance from the current center, which can be performed in $\mathcal{O}(mn)$ operations. The initial constant factor approximation can also be computed in $\mathcal{O}(mn)$ operations. Therefore, Algorithm 5.1 terminates in $\mathcal{O}(mn/(\mu^*\epsilon))$ operations. □

Next, we establish that Algorithm 5.1 computes an ϵ-core set upon termination.

Theorem 5.2. *Let $\epsilon \in (0,1)$ and let u^η denote the final iterate computed by Algorithm* 5.1. *Then, $\mathcal{X}_\eta \subset \mathcal{A}$ is an ϵ-core set of $\mathcal{A}$. Furthermore, $|\mathcal{X}_\eta| = \mathcal{O}(1/(\mu^*\epsilon))$.*

Proof. We first prove the second statement. Note that $\mathcal{X}_0$ is initialized with two elements, and each iteration adds at most one element to $\mathcal{X}_k$. Therefore, $|\mathcal{X}_\eta| = \mathcal{O}(1/(\mu^*\epsilon))$ by Lemma 5.3.

Note that the restriction of u^η to its positive components is a feasible solution of the dual formulation of the instance $(\mathcal{X}_\eta, \mathcal{W}_\eta)$ with the same objective function value γ^η, where $\mathcal{W}_\eta := \{\omega_j\colon a^j \in \mathcal{X}_\eta\}$. Therefore,

$$\gamma^\eta \le \gamma_{\mathcal{X}_\eta} \le \gamma_{\mathcal{A}} \le (1+\delta_\eta)\gamma^\eta \le (1+\epsilon)^2\gamma^\eta,$$

where $\gamma_{\mathcal{X}_\eta}$ denotes the optimal value of the dual formulation corresponding to the instance $(\mathcal{X}_\eta, \mathcal{W}_\eta)$. It follows that

$$\rho_{\mathcal{X}_\eta} \le \rho_{\mathcal{A}} \le (1+\epsilon)\rho_{\mathcal{X}_\eta},$$

where $\rho_{\mathcal{X}_\eta} = (\gamma_{\mathcal{X}_\eta})^{1/2}$, which implies that $\mathcal{X}_\eta$ is an ϵ-core set of $\mathcal{A}$. □

Note that each of the previous results depends on the parameter μ^*, which can be determined only upon the termination of Algorithm 5.1. However, this parameter can be a priori bounded below by

$$\tau = \frac{\min_{i=1,\ldots,m} \nu_i}{\max_{i=1,\ldots,m} \nu_i}, \tag{21}$$

where ν_i is defined as in (4), since each μ^k is the ratio of a convex combination of the ν_i to some ν_j. Therefore, each of the results established in Theorems 5.1 and 5.2 holds true if μ^* is replaced by τ. This implies that Algorithm 5.1 terminates in $\mathcal{O}(mn/(\tau\epsilon))$ arithmetic operations and computes an ϵ-core set of size $\mathcal{O}(1/(\tau\epsilon))$ for $\epsilon \in (0,1)$, which asymptotically matches the bound of Theorem 4.1 without the requirement to solve smaller subproblems exactly. We remark that the overall complexity of Algorithm 5.1 and the asymptotic core set size reduce to $\mathcal{O}(mn/\epsilon)$ and $\mathcal{O}(1/\epsilon)$, respectively, for the special case of the minimum enclosing ball problem since $\tau = 1$. These results match the current best-known bounds for the minimum enclosing ball problem (Yıldırım 2008, Bădoiu and Clarkson 2008).

5.2. Relation to Other Core Set Results

Recently, Clarkson (2008) studied the properties of several variants of the Frank-Wolfe algorithm for general concave maximization problems over the unit simplex, of which the dual formulation of the weighted Euclidean one-center problem is a special case. In particular, he proposed a general definition of an *additive* core set based on an additive error on the optimal value as opposed to the multiplicative one (see (3)) adopted in our setting. He derived upper bounds on the size of an additive core set for the general problem. He established that his definition of an additive core set almost coincides with the usual definition of a multiplicative core set in the special case of the dual formulation of the minimum enclosing ball problem. As such, his results imply the known bound of $\mathcal{O}(1/\epsilon)$ on the size of an ϵ-core set for this problem.

In this subsection, we discuss the relations between his bound on the size of an additive core set and our bound on the size of a multiplicative one. In particular, we establish that Clarkson's (2008) additive core set result can be transformed into a multiplicative core set result for the weighted Euclidean one-center problem. However, it turns out that these implied bounds are not asymptotically better than our bounds.

5.2.1. The Nonlinearity Measure $\mathcal{C}$.

Consider the following optimization problem:

$$\begin{aligned} \max_u \quad & \Psi(u) \\ \text{subject to} \quad & u \in \mathcal{S}, \end{aligned} \tag{22}$$

where $\Psi\colon \mathbb{R}^m \to \mathbb{R}$ is a twice differentiable concave function and $\mathcal{S} := \{u \in \mathbb{R}^m\colon e^T u = 1,\ u \ge 0\}$ is the unit simplex. Clearly, this class of problems includes the dual optimization problem $(\mathcal{D})$.

Using the Frank-Wolfe algorithm (and some of its variants), Clarkson (2008) established that, for any $\delta' > 0$, one can compute a feasible solution $u' \in \mathcal{S}$ such that

$$\Psi(u') \ge \Psi(u^*) - \delta',$$

where $u^* \in \mathcal{S}$ is an optimal solution of (22), in at most $\mathcal{O}(\mathcal{C}(\Psi)/\delta')$ iterations. Since his initial solution has only one nonzero component, u' has at most $\mathcal{O}(\mathcal{C}(\Psi)/\delta')$ positive components due to the nature of add-iterations in the Frank-Wolfe algorithm. Here, $\mathcal{C}(\Psi)$ is a measure of nonlinearity of the objective function Ψ and is defined as

$$\mathcal{C}(\Psi) := \sup_{u,\, z \in \mathcal{S},\, y = u + \alpha(u-z) \in \mathcal{S}} \frac{1}{\alpha^2}(\Psi(u) + (y-u)^T \cdot \nabla\Psi(u) - \Psi(y)). \tag{23}$$

Essentially, $\mathcal{C}(\Psi)$ is an upper bound on the (scaled) difference between the function Ψ and the linear approximation to Ψ measured over all feasible solutions. For instance, $\mathscr{C}(\Psi) = 0$ for a linear function Ψ. Therefore, $\mathscr{C}(\Psi)$ can be viewed as a measure of "flatness" of Ψ (Clarkson 2008).

Clarkson's (2008) upper bound on the size of the additive core set is useful if $\mathscr{C}(\Psi)$ can be bounded above for a given function Ψ. For instance, Clarkson showed that an upper bound on $\mathscr{C}(\Psi)$ can be easily derived if Ψ is a quadratic function, which is the case for the objective function of the dual formulation of the minimum enclosing ball problem. We now establish that $\mathscr{C}(\Phi)$ can be similarly bounded above for the objective function Φ of the problem $(\mathscr{D})$ even though Φ is not a quadratic function for the weighted problem. Recall that

$$\Phi(u) = \sum_{i=1}^{m} u_i \nu_i (a^i)^T a^i - \frac{1}{\sum_{i=1}^{m} u_i \nu_i} \left(\sum_{i=1}^{m} u_i \nu_i a^i \right)^T \left(\sum_{i=1}^{m} u_i \nu_i a^i \right).$$

It follows that

$$\nabla \Phi(u) = d - \frac{2}{u^T \nu} M u + \frac{u^T M u}{(u^T \nu)^2} \nu,$$
$$\nabla^2 \Phi(u) = -\frac{2}{(u^T \nu)^3} P(u) M P(u)^T,$$

where $d \in \mathbb{R}^m$ and is defined as $d_i := \nu_i \|a^i\|^2$, $i = 1, \ldots, m$, $\nu := [\nu_1, \ldots, \nu_m]^T$,

$$A := \left[a^1, \ldots, a^m\right], \quad M := \mathrm{Diag}(\nu) A^T A \mathrm{Diag}(\nu),$$

and

$$P(u) := \nu u^T - (u^T \nu) I.$$

By the second mean value theorem,

$$\mathscr{C}(\Phi) \le \sup_{u, z \in \mathscr{S}} -\frac{1}{2} (z - u)^T \nabla^2 \Phi(\tilde{u})(z - u),$$

where $\tilde{u} \in \mathscr{S}$ is a point that lies on the line segment from u to z. Therefore,

$$\mathscr{C}(\Phi) \le \sup_{u, z \in \mathscr{S}} \frac{1}{(\bar{u}^T \nu)^3} (z - u)^T P(\bar{u}) M P(\bar{u})^T (z - u),$$

where $\bar{u} \in \mathscr{S}$ is any point that lies on the line passing through x and z. The first term on the right-hand side can be bounded above by $1/(\min_i \nu_i)^3$. Using the fact that $\bar{u} = u + \beta(z - u)$ for some $\beta \in \mathbb{R}$, it follows that

$$\begin{aligned} \|P(\bar{u})^T (z - u)\| &= \|(\nu^T z) u - (u^T \nu) z\| \\ &\le (\nu^T z) \|u\| + (u^T \nu) \|z\| \le 2 \Big(\max_i \nu_i \Big), \end{aligned}$$

since u and z are on the unit simplex and have Euclidean norm at most one. Furthermore,

$$\|M\| \le \|A\|^2 \|\mathrm{Diag}(\nu)\|^2 = \Big(\max_i \nu_i \Big)^2 \|A\|^2,$$

where $\|\cdot\|$ denotes the operator norm of a matrix. Therefore, we obtain

$$\mathscr{C}(\Phi) \le \frac{(\max_i \nu_i)^4}{(\min_i \nu_i)^3} 4 \|A\|^2 = \frac{4 (\max_i \nu_i) \|A\|^2}{\tau^3}, \tag{24}$$

where τ is defined as in (21).

By (24), we immediately obtain an upper bound of $\mathscr{O}([(\max_i \nu_i)\|A\|^2]/\tau^3 \delta')$ on the size of a δ'-additive core set for the weighted Euclidean one-center problem.

5.2.2. Additive vs. Multiplicative Error. In this section, given a feasible solution of $(\mathscr{D})$ that has a small multiplicative (or relative) error with respect to the optimal value $\Phi(u^*)$, we establish a bound on the corresponding additive error. This will enable us to relate our bounds to those arising from Clarkson's (2008) results.

Given $\epsilon > 0$, Algorithm 5.1 computes a feasible solution $u^k \in \mathscr{S}$ such that

$$\Phi(u^k) \le \Phi(u^*) \le (1 + \delta) \Phi(u^k),$$

where $\delta \le (1 + \epsilon)^2 - 1 = \mathscr{O}(\epsilon)$. Therefore,

$$\Phi(u^k) \ge \Phi(u^*) - \delta \Phi(u^k), \tag{25}$$

which implies that (25) is satisfied with an additive error δ' if

$$\delta' \le \delta \Phi(u^k). \tag{26}$$

We now establish an upper bound on δ' independent of the function Φ to compute a lower bound on $\mathscr{C}(\Phi)/\delta'$.

Note that

$$\begin{aligned} \Phi(u^k) &\ge \Phi(u^0) \\ &= \frac{\|a^j - a^{j_*}\|^2}{(1/\omega_j + 1/\omega_{j_*})^2} \\ &\ge \frac{\|a^j - a^l\|^2}{(1/\omega_j + 1/\omega_l)^2} \\ &= (1/4) \Big(\min_i \nu_i \Big) \|a^j - a^l\|^2, \end{aligned}$$

where j and j^* are defined as in Algorithm 3.1 and $a^l \in \mathscr{A}$ is the point with the largest Euclidean distance from a^j. It follows that (26) is satisfied if

$$\delta' \le (1/4) \delta \Big(\min_i \nu_i \Big) \|a^j - a^l\|^2. \tag{27}$$

We remark that the inequality (27) that establishes the relation between δ and δ' is asymptotically tight as illustrated by the following example. Let $\mathscr{A} = \{-1, 0, 1\}$ and $\mathscr{W} = \{1, 1 + \kappa, 1\}$, where $\kappa > 0$. It is easy to verify that $\Phi(u^0) = [(1 + \kappa)/(2 + \kappa)]^2$ and $\Phi(u^*) = 1$. Clearly, $\Phi(u^*) \le (1 + \delta)\Phi(u^0)$ with $\delta = [1/(1 + \kappa)](2 + 1/(1 + \kappa))$ and $\Phi(u^0) \ge \Phi(u^*) - \delta'$ with $\delta' = (2\kappa + 3)/(2 + \kappa)^2$. Therefore, both δ' and the right-hand side of (27) tend to 3/4 as κ goes to zero.

Next, we establish a lower bound on $\mathscr{C}(\Phi)$. Recall that

$$\mathscr{C}(\Phi) := \sup_{u,\, z\in\mathscr{S},\, y=(1-\alpha)u+\alpha z\in\mathscr{S}} \frac{1}{\alpha^2}(\Phi(u)+(y-u)^T \cdot \nabla\Phi(u) - \Phi(y)),$$

which implies that any feasible choices of $u, y, z \in \mathscr{S}$ will yield a lower bound on $\mathscr{C}(\Phi)$. Let

$$u = e^l, \quad z = e^j, \quad y = z = e^j \quad (\alpha = 1),$$

where the indices j and l are chosen such that $a^j \in \mathscr{A}$ is the point with the largest weight ω_j and $a^l \in \mathscr{A}$ is the point with the largest Euclidean distance from a^j. With these choices, we have $\Phi(u) = \Phi(y) = 0$. Hence,

$$\begin{aligned}\mathscr{C}(\Phi) &\geq (e^j - e^l)^T (d - (2/\nu_l)Me^l + (M_{ll}/(\nu_l)^2)\nu)\\ &= \nu_j\|a^j\|^2 - \nu_l\|a^l\|^2 - (2/\nu_l)M_{jl}\\ &\quad + (2/\nu_l)M_{ll} + (M_{ll}\nu_j)/(\nu_l)^2 - M_{ll}/\nu_l\\ &= \nu_j\|a^j\|^2 - \nu_l\|a^l\|^2 - 2\nu_j(a^j)^T(a^l) + \nu_l\|a^l\|^2 + \nu_j\|a^l\|^2\\ &= \nu_j\|a^j - a^l\|^2,\end{aligned}$$

where we used the fact that $M_{ik} = \nu_i\nu_k(a^i)^T(a^k)$. Therefore,

$$\mathscr{C}(\Phi) \geq \nu_j\|a^j - a^l\|^2. \tag{28}$$

Combining (27) with (28), it follows that

$$\frac{\mathscr{C}(\Phi)}{\delta'} \geq \frac{4}{\tau\delta},$$

which implies that Clarkson's (2008) result does not improve our upper bound of $\mathscr{O}(1/\tau\epsilon) = \mathscr{O}(1/\tau\delta)$ on the size of an ϵ-core set, even if a matching upper bound for $\mathscr{C}(\Phi)$ could be found.

We remark that Clarkson's (2008) analysis is quite general, and some of his results yield the tightest possible bounds on the size of core sets as in the case of the minimum enclosing ball problem. However, for specific problems such as the problem considered in this paper, our line of analysis may lead to core set bounds that are at least as good as the ones implied by his results. Furthermore, as pointed out in Clarkson (2008), there are certain problems of the form (22) with objective functions Ψ for which $\mathscr{C}(\Psi)$ is unbounded. For instance, the objective function of the dual formulation of the minimum enclosing ellipsoid problem satisfies this property. For such problems, bounds that depend on $\mathscr{C}(\Psi)$ are not useful, whereas the line of analysis adopted in this paper may still yield small core set results (Kumar and Yıldırım 2005, Todd and Yıldırım 2007). These observations seem to suggest that problem-specific approaches, although narrower in scope, may lead to sharper bounds than a general-purpose approach with a much wider scope.

6. Computational Experiments

In this section, we present and discuss our computational results. We implemented Algorithm 5.1 in MATLAB and conducted our computational experiments on input sets generated randomly using various distributions. Specifically, we considered the following two classes of input sets:

1. Normal distribution: Each coordinate of each input point was generated using the standard normal distribution.
2. Uniform distribution: Each coordinate of each input point was generated using the uniform distribution on the interval $(0, 1)$.

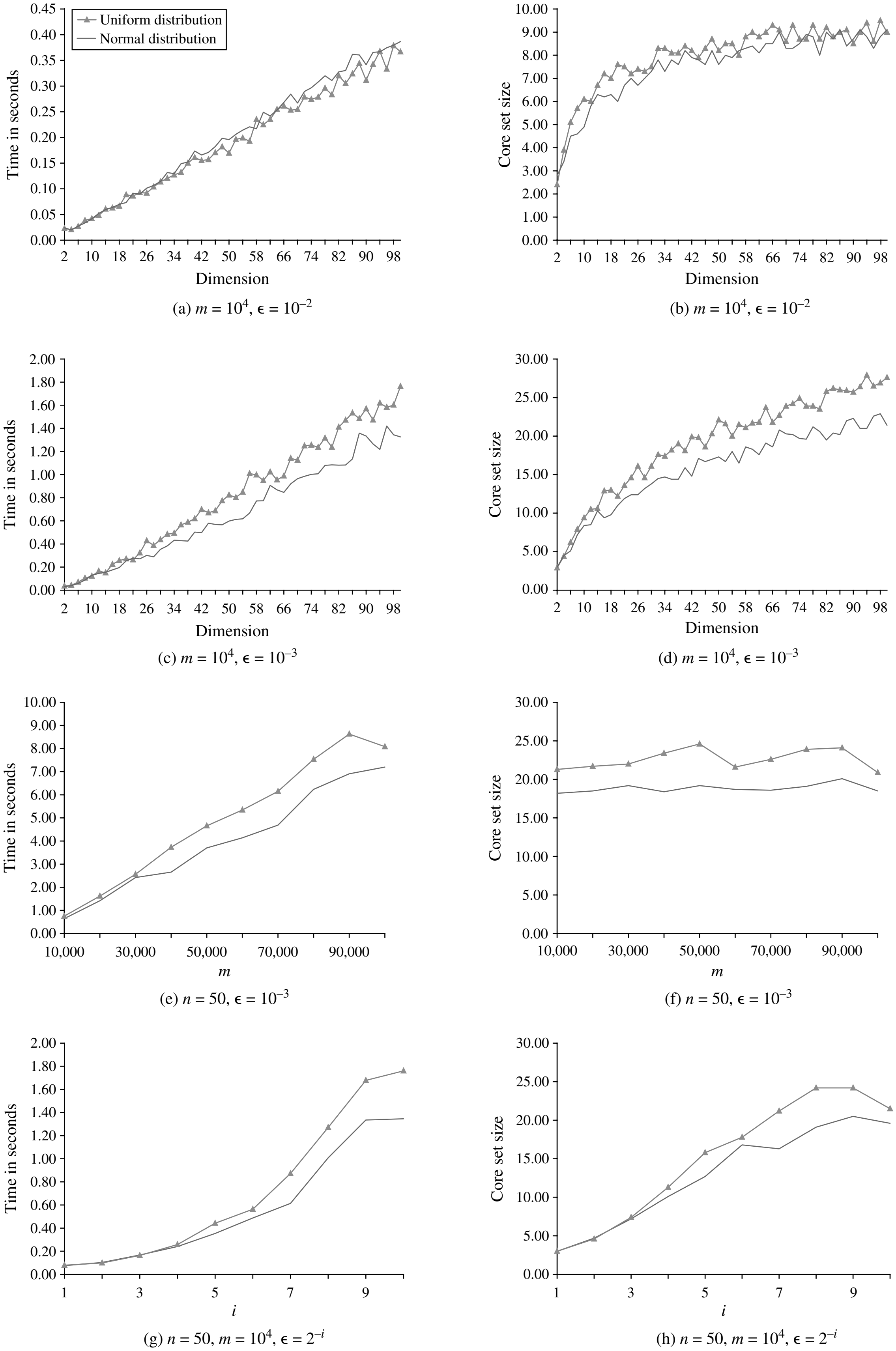
For each input point, the corresponding weight was chosen uniformly from the interval $(0, 1)$. Our experiments were performed on a notebook computer with an Intel Core 2 CPU T7400 2.17 GHz processor, 2 GB of RAM, and a 120 GB, 5,400 rpm hard drive.

Our first experiment provides information about the performance of Algorithm 5.1 on instances of the weighted Euclidean one-center problem in small dimensions (see Table 1). For each instance, the number of points m was set at 1,000. All points were uniformly generated from the n-dimensional unit cube. We used $\epsilon = 10^{-4}$ in our experiments. Table 1 reports, for each dimension n, the core set sizes, CPU times, number of iterations, value of τ defined by (21), and μ^* as defined in Corollary 5.1 averaged over 50 runs.

Table 1 reveals that Algorithm 5.1 is capable of quickly computing a highly accurate solution in small dimensions. In particular, the sizes of core sets computed by the algorithm are significantly smaller than the worst-case theoretical estimate. Furthermore, the sizes of core sets are also considerably smaller than the number of iterations, which suggests that drop-iterations may be effective in maintaining small core sets. Next, the values of τ are much smaller than the values of μ^*, which implies that τ can be a rather loose lower bound on μ^*. Therefore, the expression of the complexity results in terms of τ seems to be a gross overestimate at least for the experimental setup used in Table 1. Finally, we remark that Drezner and Gavish (1985) used essentially the same

Table 1 **Computational Results with Uniform Distribution for $m = 1{,}000$**

n	$\lvert\mathscr{X}\rvert$	Time (sec)	Iterations	τ	μ^*
2	2.480	0.012	44.120	1.485×10^{-9}	0.906
3	3.360	0.012	53.880	0.159×10^{-9}	0.903
4	4.320	0.014	56.380	7.754×10^{-9}	0.897
5	4.820	0.017	69.720	2.387×10^{-9}	0.906
6	5.760	0.015	61.280	9.853×10^{-9}	0.907
7	6.140	0.016	61.800	1.423×10^{-9}	0.908
8	6.760	0.019	72.960	2.770×10^{-9}	0.895
9	7.240	0.018	67.240	0.029×10^{-9}	0.918
10	7.660	0.020	75.020	1.689×10^{-9}	0.915



(a) $m = 10^4$, $\epsilon = 10^{-2}$

(b) $m = 10^4$, $\epsilon = 10^{-2}$

(c) $m = 10^4$, $\epsilon = 10^{-3}$

(d) $m = 10^4$, $\epsilon = 10^{-3}$

(e) $n = 50$, $\epsilon = 10^{-3}$

(f) $n = 50$, $\epsilon = 10^{-3}$

(g) $n = 50$, $m = 10^4$, $\epsilon = 2^{-i}$

(h) $n = 50$, $m = 10^4$, $\epsilon = 2^{-i}$

Figure 1 Various Experimental Results from the Implementation of Our Algorithm

experimental settings in the implementation of their ellipsoid algorithm. The number of iterations reported in Table 1 are noticeably smaller than their counterparts in their implementation. We find it remarkable that a simple first-order algorithm can outperform a polynomial-time algorithm in terms of the total number of iterations.

Figure 1 presents several graphs in an attempt to provide further insights into the performance of Algorithm 5.1 for larger instances of the weighted Euclidean one-center problem. In particular, we aim to establish how the practical performance is affected by input parameters such as the number of points m, the dimension n, and the accuracy ϵ using two classes of input sets.

Figure 1 is organized such that the vertical axis denotes the CPU time on each graph in the first column and it corresponds to the size of the core set on each graph in the second column. In each row, exactly two of the three parameters m, n, and ϵ are fixed and the third one is varied, which is denoted in the horizontal axis. All of the data points in Figure 1 were generated by averaging the CPU times and core set sizes over 10 runs.

Figures 1(a) and 1(b) present the CPU times of Algorithm 5.1 and the sizes of core sets returned by the algorithm, respectively, for different values of the dimension n using $m = 10^4$ and $\epsilon = 0.01$. Figures 1(c) and 1(d) present the corresponding results with the same experimental setup except for the choice of 0.001 for ϵ. A study of these graphs reveals that the CPU times tend to increase linearly with the dimension n for both classes of input sets as predicted by the theoretical complexity results. The sizes of core sets seem to exhibit a weaker dependence on n but are usually much smaller than the corresponding worst-case estimate of $\mathcal{O}(1/(\tau\epsilon))$. Clearly, both the CPU times and the sizes of core sets increase as ϵ decreases.

The CPU times and the sizes of core sets with different numbers of input points m are presented in Figures 1(e) and 1(f) using $n = 50$ and $\epsilon = 0.001$. Once again, the CPU times seem to exhibit a linear dependence on m for each class of input sets. On the other hand, there seems to be no relation between the size of the core set and the number of input points.

Finally, Figures 1(g) and 1(h) illustrate the relationship between the performance of Algorithm 5.1 and the accuracy ϵ using $n = 50$ and $m = 10^4$. The accuracy ϵ is chosen as smaller powers of 2. Note that the horizontal axis has a logarithmic scale. As predicted by our theoretical analysis, both the CPU times and core set sizes increase as ϵ decreases for both classes of input sets. It is worth noting that core set sizes tend to increase linearly with respect to $\log(1/\epsilon)$, which contributes to the efficiency of Algorithm 5.1 in practice.

In conclusion, Algorithm 5.1 seems to be able to compute an approximate solution for larger instances of the weighted Euclidean one-center problem in a very reasonable amount of time. We remark that the CPU times are remarkably small. In particular, an instance of the problem with 10^6 input points in 50 dimensions can be solved in under 10 seconds. Also notable is the fact that core set sizes tend to be quite small, which is one of the main driving forces behind the practical efficiency of Algorithm 5.1.

7. Concluding Remarks

In this paper, we developed and analyzed an algorithm for the weighted Euclidean one-center problem. Our algorithm explicitly computes a core set whose size is independent of the number of points and the dimension. Our results subsume and extend some of the previously known results for the minimum enclosing ball problem to a larger class of problems. Computational experiments reveal the efficiency of our algorithm in practice.

In the near future, we intend to work on different variants of this problem such as the weighted Euclidean one-center problem with outliers. Another interesting research direction would be the investigation of the tightest bounds on the size of core sets.

Acknowledgments

The first author was partially supported by the National Science Foundation through CAREER Grant CCF-0643593 and the second author by TÜBİTAK (Turkish Scientific and Technological Research Council) Grant 107M411 and by the Bilkent University Faculty Development Grant. The authors are grateful to the area editor, associate editor, and three anonymous referees for their valuable and constructive suggestions. The authors especially acknowledge an anonymous referee, whose comments prompted the addition of §4.

References

Agarwal, P., S. Har-Peled, K. Varadarajan. 2005. Geometric approximations via coresets. J. E. Goodman, J. Pach, E. Wezl, eds. *Combinatorial and Computational Geometry*, Vol. 52. Cambridge University Press, New York, 1–30.

Bădoiu, M., K. L. Clarkson. 2003. Smaller core-sets for balls. *Proc. 14th Annual Sympos. Discrete Algorithms, Baltimore*, SIAM, Philadelphia, 801–802.

Bădoiu, M., K. L. Clarkson. 2008. Optimal core-sets for balls. *Computational Geometry: Theory Appl.* **40**(1) 14–22.

Bădoiu, M., S. Har-Peled, P. Indyk. 2002. Approximate clustering via core-sets. *Proc. 34th Annual ACM Sympos. Theory Comput., Quebec*, ACM, New York, 250–257.

Blum, L., M. Shub, S. Smale. 1989. On a theory of computation and complexity over the real numbers: *NP*-completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc.* **21**(1) 1–46.

Chandrasekaran, R. 1982. The weighted Euclidean 1-center problem. *Oper. Res. Lett.* **1**(3) 111–112.

Clarkson, K. L. 2008. Coresets, sparse greeedy approximation, and the Franke-Wolfe algorithm. *Proc. 19th Ann. ACM-SIAM Sympos. Discrete Algorithms, San Francisco*, 922–931.

Drezner, Z., B. Gavish. 1985. ε-Approximations for multidimensional weighted location problems. *Oper. Res.* **33**(4) 772–783.

Dyer, M. E. 1986. On a multidimensional search technique and its application to the Euclidean one-centre problem. *SIAM J. Comput.* **15**(3) 725–738.

Francis, R. L. 1967. Some aspects of a minimax location problem. *Oper. Res.* **15**(6) 1163–1169.

Frank, M., P. Wolfe. 1956. An algorithm for quadratic programming. *Naval Res. Logist. Quart.* **3**(1–2) 95–110.

Grötschel, M., L. Lovász, A. Schrijver. 1988. *Geometric Algorithms and Combinatorial Optimization, Algorithms and Combinatorics*, Vol. 2. Springer-Verlag, Berlin.

Hansen, P., D. Peeters, D. Richard, J.-F. Thisse. 1985. The minisum and minimax location problems revisited. *Oper. Res.* **33**(6) 1251–1265.

Hearn, D. W., J. Vijay. 1982. Efficient algorithms for the (weighted) minimum circle problem. *Oper. Res.* **30**(4) 777–795.

Kumar, P., E. A. Yıldırım. 2005. Minimum-volume enclosing ellipsoids and core sets. *J. Optim. Theory Appl.* **126**(1) 1–21.

Kumar, P., J. S. B. Mitchell, E. A. Yıldırım. 2003. Approximate minimum enclosing balls in high dimensions using core-sets. *ACM J. Experiment. Algorithmics* **8**, Article 1.1.

Megiddo, N. 1983. The weighted Euclidean 1-center problem. *Math. Oper. Res.* **8**(4) 498–504.

Megiddo, N. 1989. On the ball spanned by balls. *Discrete Comput. Geometry* **4**(6) 605–610.

Todd, M. J., E. A. Yıldırım. 2007. On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Appl. Math.* **155**(13) 1731–1744.

Tsang, I., J. Kwok, P.-M. Cheung. 2005. Very large SVM training using core vector machines. R. G. Cowell, Z. Ghahramani, eds. *Proc. 10th Internat. Workshop on Artificial Intelligence Statist., Barbados*, Society for Artificial Intelligence and Statistics, Hoboken, NJ, 349–356. http://www.gatsby.ucl.ac.uk/aistats/fullpapers/172.pdf.

Wolfe, P. 1970. Convergence theory in nonlinear programming. J. Abadie, ed. *Integer and Nonlinear Programming*. North-Holland, Amsterdam, 1–36.

Yıldırım, E. A. 2008. Two algorithms for the minimum enclosing ball problem. *SIAM J. Optim.* **19**(3) 1368–1391.

Zhou, G., K. C. Toh, B. Sun. 2005. Efficient algorithms for the smallest enclosing ball problem. *Computational Optim. Appl.* **30**(2) 147–160.