Approximate Minimum Volume Enclosing Ellipsoids Using Core Sets

Piyush Kumar∗ E. Alper Yildirim†

April 21, 2003

Abstract

We study the problem of computing the minimum volume enclosing ellipsoid containing a given point set \( S = \{p_1, p_2, \ldots, p_n\} \subseteq \mathbb{R}^d \). Using “core sets” and a column generation approach, we develop a \((1 + \epsilon)\)-approximation algorithm. We prove the existence of a core set \( X \subseteq S \) of size at most \(|X| = \alpha = O(d \log d + \frac{1}{\epsilon^2})\). We describe an algorithm that computes the set \( X \) and a \((1 + \epsilon)\)-approximation to the minimum volume enclosing ellipsoid of \( S \) in \( O(nd^2 \alpha + \frac{\alpha^4}{\epsilon} \log \alpha) \) operations by using Khachiyan’s algorithm to solve each subproblem. This result is an improvement over the previously known algorithms especially for input sets with \( n \gg d \) and reasonably small values of \( \epsilon \). We also discuss extensions to the cases in which the input set consists of balls or ellipsoids.

1 Introduction

We study the problem of computing the minimum volume enclosing ellipsoid (MVEE) of a given point set \( S = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^d \), denoted by MVEE(\(S\)), also known as the Löwner ellipsoid for \( S \). Minimum volume enclosing ellipsoids play an important role in several diverse applications such as optimal design [29, 32], computational geometry [36, 9, 3], integer programming [21, 13], computer graphics [11, 6], pattern recognition [12], statistics [28], and nondifferentiable convex optimization [31]. Variations of this problem such as MVEE with outliers [24] have many other applications [10, 15, 19].

A (full-dimensional) ellipsoid \( E_{Q,c} \) in \( \mathbb{R}^d \) is specified by a \( d \times d \) symmetric positive definite matrix \( Q \) and a center \( c \in \mathbb{R}^d \) and is defined as

\[
E_{Q,c} = \{ x \in \mathbb{R}^d : (x - c)^T Q (x - c) \leq 1 \}. \tag{1}
\]

The volume of an ellipsoid \( E_{Q,c} \), denoted by \( \text{vol} E_{Q,c} \), is given by \( \text{vol} E_{Q,c} = B \det Q^{-\frac{1}{2}} \), where \( B \) is the volume of the unit ball in \( \mathbb{R}^d \) [13].

F. John [14] proved that MVEE(\(S\)) satisfies

\[
\frac{1}{d} \text{MVEE}(S) \subseteq \text{conv}(S) \subseteq \text{MVEE}(S), \tag{2}
\]

where \( \text{conv}(S) \) denotes the convex hull of \( S \) and the ellipsoid on the left-hand side is obtained by scaling MVEE(\(S\)) around its center by a factor of \( \frac{1}{d} \). Therefore, if \( S \) is viewed as the set of vertices of a full-dimensional polytope \( P \subseteq \mathbb{R}^d \), then MVEE(\(S\)) yields a rounded approximation of \( P \). This is a crucial ingredient in Khachiyan’s ellipsoid algorithm [16], the first polynomial-time algorithm for linear programming.

Since the work of John [14] in 1940s, several algorithms have been developed for the MVEE problem. These algorithms include the gradient-descent type first-order algorithms [32, 33, 28, 17] and interior-point

∗Department of Computer Science, Stony Brook University, Stony Brook, NY, 11794, USA. This author was supported in part by NSF through grant CCR-0098172.
†Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY, 11794 – 3600, USA. This author was supported in part by NSF through CAREER grant DMI-0237415.
algorithms [18, 26, 25, 37, 2, 38, 30]. For small dimensions \( d \), the MVEE problem can be solved in \( \mathcal{O}(d^{O(d)}n) \) operations using randomized [23, 36] or deterministic [9] algorithms.

The MVEE problem can also be solved by algorithms developed more generally for maximum determinant problems by Vandenberghe et. al. [35] and Toh [34]. Several algorithms are compared in [38] for the closely related problem of computing a maximum volume inscribed ellipsoid (MVIE) in a polytope (Khachiyan and Todd [18] prove the equivalence between the MVIE and the MVEE problems for appropriate representations of polytopes). Sun and Freund [30] propose a practical algorithm that uses an active set strategy to solve large-scale instances of the MVEE problem.

For a given set \( S \) of \( n \) points in \( \mathbb{R}^d \) and \( \delta > 0 \), Khachiyan’s algorithm [17] computes an ellipsoid \( E \supseteq S \) such that \( \text{vol}(E) \leq (1 + \delta) \text{vol(MVEE}(S)) \) in \( \mathcal{O}(n^{3.5} \log(\frac{n}{\delta})) \) operations in the real number model of computation. This is currently the best known complexity result when \( d \) is part of the input.

An ellipsoid \( E_{Q,c} \supseteq S \) is said to be a \((1 + \epsilon)\)-approximation to the MVEE\((S)\) if \( \text{vol}(E_{Q,c}) \leq \text{vol}(1 + \epsilon) \text{MVEE}(S) \). It follows that \( \text{vol}(E_{Q,c}) \) is a \((1 + \epsilon)^d\)-approximation to \( \text{vol} \text{MVEE}(S) \). A set \( X \subseteq S \) is called a core set for \( S \) [8, 7, 20] if \((1 + \epsilon)\text{MVEE}(X)\) is a \((1 + \epsilon)\)-approximation to the MVEE\((S)\). It follows from (1) that

\[
(1 + \epsilon)E_{Q,c} := \{ x \in \mathbb{R}^d : (x - c)^TQ(x - c) \leq (1 + \epsilon)^2 \}.
\]

In this paper, we develop an algorithm that computes a \((1 + \epsilon)\)-approximation to the MVEE\((S)\) for a given point set \( S \subseteq \mathbb{R}^d \). We establish the existence of a core set \( X \subseteq S \) with the property that \( |X| = \mathcal{O}(d (\log d + \frac{1}{\epsilon})) \). Combining this result with a column generation strategy and using Khachiyan’s algorithm [17] to solve each subproblem, we prove that an \((1 + \epsilon)\)-approximation to the MVEE problem can be computed in \( \mathcal{O}(n^d \alpha + \alpha^{4.5} \log \frac{1}{\epsilon}) \) operations, where \( \alpha = \mathcal{O}(d (\log d + \frac{1}{\epsilon})) \) denotes the size of the core set \( X \). Compared with Khachiyan’s algorithm [17], our algorithm enjoys a better complexity result for large-scale instances with \( n \gg d \) and reasonably small values of \( \epsilon \).

This paper extends the previous work of Kumar, Mitchell and Yıldırım [20] on the minimum enclosing ball problem in higher dimensions. The authors developed an algorithm that computes a \((1 + \epsilon)\)-approximation to the minimum enclosing ball of a given set of points based on the existence of a core set of size \( \mathcal{O} (\frac{n}{\epsilon}) \). The computational results indicated that the algorithm can solve large instances of the minimum enclosing ball problem in higher dimensions significantly faster than the previously known approaches. Furthermore, the core set sizes in practice tend to be significantly smaller than the theoreitical worst-case bound, enhancing the effectiveness of the algorithm in practice. In this paper, we extend a similar approach to the more general problem of computing an approximate MVEE of a given point set. Despite the similarity of the underlying approach, the theoretical analysis uses entirely different tools.

The paper is organized as follows. In Section 2, we review formulations of the MVEE problem as an optimization problem. We develop the main core set result and describe an approximation algorithm in Section 3. We discuss extensions of our approach to the cases when the input set \( S \) consists of balls or ellipsoids and conclude the paper in Section 4.

## 2 Formulations as an Optimization Problem

In this section, we discuss formulations of the MVEE problem as an optimization problem. Throughout the rest of this paper, we make the following assumption, which guarantees that the minimum enclosing ellipsoid will be full-dimensional.

**Assumption 2.1** The points \( p_1, \ldots, p_n \) are affinely independent.

The MVEE problem can be formulated as an optimization problem in several different ways (see, e.g., [30]). We consider two formulations in this section. Given a set \( S \subseteq \mathbb{R}^d \) of \( n \) points \( p_1, \ldots, p_n \), MVEE\((S)\) can be computed by lifting each point to \((d+1)\)-dimensional space by appending 1 as the \((d+1)\)st component to each of the \( n \) points, finding the minimum volume enclosing ellipsoid \( E' \subseteq \mathbb{R}^{d+1} \) of the lifted point set centered at the origin and intersecting \( E' \) by the \( d \)-dimensional hyperplane consisting of all points in \( \mathbb{R}^{d+1} \).

\(^1\)Log denotes the natural logarithm throughout the paper.
whose \((d + 1)\)st component is given by 1 (see, e.g., [18, 26]). This observation gives rise to the following convex optimization problem to compute \(\text{MVEE}(S)\):

\[
\begin{align*}
(P(S)) \quad & \min_{M} \quad -\frac{1}{2} \log \det M \\
& \text{s.t.} \quad \begin{bmatrix} p_i^T & 1 \end{bmatrix} M \begin{bmatrix} p_i \\ 1 \end{bmatrix} \leq 1, \quad i = 1, \ldots, n, \\
& \quad M \in \mathbb{R}^{(d+1) \times (d+1)} \text{ is symmetric and positive definite}.
\end{align*}
\]

The Lagrangian dual of \((P(S))\) is equivalent to

\[
(D(S)) \quad \max_u \quad \log \det V(u) := \log \det \left( \sum_{i=1}^n u_i \begin{bmatrix} p_i \\ 1 \end{bmatrix} \begin{bmatrix} p_i^T & 1 \end{bmatrix} \right) \\
\text{s.t.} \quad e^T u = 1, \\
& \quad u \geq 0,
\]

where \(u \in \mathbb{R}^n\) is the decision variable and \(e \in \mathbb{R}^n\) denotes the vector of all ones.

Since \((D(S))\) is a concave optimization problem, \(u^* \in \mathbb{R}^n\) is an optimal solution (along with dual solutions \(s^* \in \mathbb{R}^n\) and \(\lambda^* \in \mathbb{R}\)) if and only if the following optimality conditions are satisfied:

\[
\begin{align*}
[p_i^T & 1] V(u^*)^{-1} \begin{bmatrix} p_i \\ 1 \end{bmatrix} + s_i^* = \lambda^*, \quad i = 1, \ldots, n \quad (4) \\
e^T u^* &= 1, \quad (5) \\
u_i^* s_i^* &= 0, \quad i = 1, \ldots, n, \quad (6)
\end{align*}
\]
together with \(u^* \geq 0\) and \(s^* \geq 0\). Multiplying both sides of the first equation by \(u_i^*\) and summing up for \(i = 1, \ldots, n\) yields \(\lambda^* = d + 1\). Consequently, \(\frac{1}{d+1} V(u^*)^{-1}\) along with Lagrange multipliers \(\frac{d+1}{2} u^*\) satisfy the necessary and sufficient optimality conditions for \((P(S))\).

It follows from this observation that an optimal solution \(u^*\) for \((D(S))\) can be used to compute \(\text{MVEE}(S)\). Since \(\frac{1}{d+1} V(u^*)^{-1}\) is an optimal solution for \((P(S))\), \(\text{MVEE}(S)\) is given by the intersection of the minimum enclosing ellipsoid of the lifted point set with the hyperplane \(H := \left\{ \begin{bmatrix} x \\ 1 \end{bmatrix} : x \in \mathbb{R}^d \right\} \):

\[
\text{MVEE}(S) = \{ x \in \mathbb{R}^d : [x^T \\ 1] \frac{1}{d+1} V(u^*)^{-1} \begin{bmatrix} x \\ 1 \end{bmatrix} \leq 1 \}.
\]

(7)

Let \(P \in \mathbb{R}^{d \times n}\) be the matrix whose \(i\)th column is given by \(p_i\). It follows that

\[
V(u^*) = \begin{bmatrix} P \text{diag}(u^*) P^T & P u^* \\ (P u^*)^T & 1 \end{bmatrix} = \begin{bmatrix} I & P u^* \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P \text{diag}(u^*) P^T - P u^* (P u^*)^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ (P u^*)^T & 1 \end{bmatrix},
\]

(8)

where \(\text{diag}(u^*)\) denotes the \(d \times d\) diagonal matrix with \(u^*\) as its diagonal entries and \(I\) denotes the \(d \times d\) identity matrix. Inverting both sides in (8) yields

\[
V(u^*)^{-1} = \begin{bmatrix} I & 0 \\ -(P u^*)^T & 1 \end{bmatrix} \begin{bmatrix} (P \text{diag}(u^*) P^T - P u^* (P u^*)^T)^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -P u^* \\ 0 & 1 \end{bmatrix},
\]

(9)

Substituting (9) in (7) gives

\[
\text{MVEE}(S) = E_{Q,c} := \{ x \in \mathbb{R}^d : (x - c)^T Q (x - c) \leq 1 \},
\]

(10)

where

\[
Q := \frac{1}{d} \left( P \text{diag}(u^*) P^T - P u^* (P u^*)^T \right)^{-1}, \quad c := P u^*.
\]

(11)

This establishes the following result.
Lemma 2.1 Let \( u^* \in \mathbb{R}^n \) be an optimal solution to \((D(S))\) and let \( P \in \mathbb{R}^{d \times n} \) be the matrix whose \( i \)th column is given by \( p_i \). Then, \( \text{MVEE}(S) = E_{Q,c} \), where \( Q \in \mathbb{R}^{d \times d} \) and \( c \in \mathbb{R}^d \) are given by (11). Furthermore,

\[
\log \text{vol MVEE}(S) = \log B + \frac{d}{2} \log d + \frac{1}{2} \log \det V(u^*),
\]

where \( B \) is the volume of the unit ball in \( \mathbb{R}^d \).

**Proof.** We only need to prove the second part. Note that \( \text{vol MVEE}(S) = B \det Q^{-\frac{1}{2}} \), where \( Q \) is defined as in (11). Therefore,

\[
\log \text{vol MVEE}(S) = \log B + \frac{d}{2} \log d + \frac{1}{2} \log \det (P \text{diag}(u^*)P^T - P u^*(P u^*)^T).
\]

It follows from (8) that \( \log \det V(u^*) = \log \det (P \text{diag}(u^*)P^T - P u^*(P u^*)^T) \), establishing (12). \qed

3 Core Sets and an Approximation Algorithm

In this section, we develop our main core set result and describe an approximation algorithm for the MVEE problem based on this result.

**Algorithm 1** Outputs a \((1 + \epsilon)\)-approximation of \( \text{MVEE}(S) \) and a Core Set \( X \)

**Require:** Input set of points \( S = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^d, \epsilon, X_0 \subseteq S \)

1: \( X \leftarrow X_0 \)
2: \( \text{loop} \)
3: Compute \( \text{MVEE}(X) = E_{Q,c} \) using Khachiyan’s algorithm.
4: if \( S \setminus (1 + \epsilon)E_{Q,c} = \emptyset \) then
5: Return \( \text{MVEE}(X), X \)
6: else
7: \( j \leftarrow \arg \max_{i=1, \ldots, n} (c - p_i)^T Q(c - p_i) \).
8: end if
9: \( X \leftarrow X \cup \{p_j\} \)
10: \( \text{end loop} \)

The algorithm, outlined above, is based on a column generation idea. In Section 3.1, we show the existence of a small subset \( X_0 \subseteq S \) such that \( \text{vol MVEE}(X_0) \) is a provable approximation to \( \text{vol MVEE}(S) \). This set will constitute the initial core set \( X_0 \) in Algorithm 1. Section 3.2 establishes that each core set update (cf. line 9 in Algorithm 1) is guaranteed to increase the volume of the enclosing ellipsoid by a function of \( \epsilon \). Finally, we discuss the size of the core set at the termination of Algorithm 1 and provide a worst-case complexity analysis in Section 3.3.

3.1 Core Set Initialization

The next lemma establishes the initial core set result.

Lemma 3.1 There exists a subset \( X_0 \subseteq S \) with \( |X_0| = 2d \) such that \( \text{vol MVEE}(S) \leq O(d^d) \text{ vol MVEE}(X_0) \). The set \( X_0 \) can be calculated in \( O(nd^2) \) time.

**Proof.** In order to construct \( X_0 \), we invoke the algorithm of [5, 22]. The algorithm starts by picking an arbitrary direction \( b_1 \in \mathbb{R}^d \) and finds the supporting hyperplanes \( H_1^+ \) and \( H_1^- \) to the convex hull of \( S \) in the directions of \( b_1 \) and \(-b_1\), which can be performed in \( O(nd) \) operations. Let \( p_1^+ \) and \( p_1^- \) be the contact points of \( H_1^+ \) and \( H_1^- \) with the convex hull of \( S \). The points \( p_1^+ \) and \( p_1^- \) are added to \( X_0 \). The next direction is chosen randomly from the set of vectors perpendicular to the affine hull of \( \{p_1^+, p_1^-\} \) and
the same procedure is repeated \( d \) times in \( \mathcal{O}(nd^2) \) operations, after which \( |X_0| = 2d \). From [5], it follows that \( \text{vol conv}(S) \leq d\text{vol conv}(X_0) \). Combining this with (2), we have \( \frac{1}{d^2} \text{vol MVEE}(S) \leq \text{vol conv}(S) \leq d\text{vol conv}(X) \leq d\text{vol MVEE}(X_0) \). This implies that \( \text{vol MVEE}(S) \leq d^d \text{vol MVEE}(X_0) \). Note that this lemma extends to the case when \( S \) is a collection of balls or ellipsoids. In the general case, however, the cost of computing \( X_0 \) may be higher (see Lemma 5.1 in the Appendix).

\[\]

### 3.2 Core Set Updates

We next discuss core set updates. Let \( X \subseteq S \) be the current core set. Algorithm 1 computes MVEE(\( X \)) using Khachiyan’s algorithm [17] (cf. line 3). If the stopping condition on line 4 is satisfied, then \( X \) is a core set and \((1 + \epsilon)\text{MVEE}(X)\) is a \((1 + \epsilon)\)-approximation to MVEE(\( S \)). Otherwise, there exists a point \( p_j \in S \) such that \( p_j \notin (1 + \epsilon)\text{MVEE}(X) \). Let \( X' := X \cup \{p_j\} \). Our goal in this subsection is to establish that \( \text{vol MVEE}(X') \geq (1 + \theta)\text{vol MVEE}(X) \), for an appropriate constant \( \theta > 0 \).

Let \( u^*_X \in \mathbb{R}^{|X|} \) be an optimal solution of (D(\( X \))) defining MVEE(\( X \)) via Lemma 2.1. Note that (D(\( X \))) is equivalent to (D(\( S \))) if one imposes the additional constraints that \( u_j = 0 \) for \( j \notin X \). In light of this observation, \( u^*_X \) can be extended appropriately to yield a feasible solution for D(\( S \)), which we denote by \( u(X) \). Since \( p_j \notin (1 + \epsilon)\text{MVEE}(X) \), it follows from Lemma 2.1 and (3) that

\[
[p_j - Pu(X)]^T \frac{1}{d} (P\text{diag}(u(X))(PT - Pu(X))(P\text{diag}(u(X)))^T)^{-1} [p_j - Pu(X)] > (1 + \epsilon)^2.
\] (14)

We define \( \gamma := 2\epsilon + \epsilon^2 \)

so that \((1 + \gamma) = (1 + \epsilon)^2 \). It follows from (9) that

\[
\frac{1}{d}[p_j^T \quad 1]V(u(X))^{-1} \begin{bmatrix} p_j \\ 1 \end{bmatrix} = [p_j - Pu(X)]^T \frac{1}{d} (P\text{diag}(u(X))(PT - Pu(X))u(X)^TP)^{-1} [p_j - Pu(X)] + \frac{1}{d}.
\]

Therefore, (14) can be rewritten as

\[
[p_j^T \quad 1]V(u(X))^{-1} \begin{bmatrix} p_j \\ 1 \end{bmatrix} > d + 1 + \gamma d,
\] (16)

where we used (15). Let \( e_j \in \mathbb{R}^n \) denote the unit vector whose \( j \)th component is 1. We next show that there exists \( \beta > 0 \) such that the feasible solution to D(\( S \)) given by \((1 - \beta)u(X) + \beta e_j \) strictly improves on the objective function.

**Lemma 3.2** Let \( \gamma > 0 \). Under the assumption (16), there exists \( \beta^* \in (0, 1) \) such that

\[
\log \det V((1 - \beta^*)u(X) + \beta^* e_j) \geq \log \det V(u(X)) + \log(1 + \gamma') - \frac{\gamma'}{1 + \gamma'}
\] (17)

where

\[
\gamma' := \frac{d}{d + 1} \gamma.
\] (18)

**Proof.** Let

\[
\rho := [p_j^T \quad 1]V(u(X))^{-1} \begin{bmatrix} p_j \\ 1 \end{bmatrix}.
\]

Our proof mimics Khachiyan’s argument in [17, Lemma 3]. Note that

\[
V((1 - \beta)u(X) + \beta e_j) = V(u(X))[I - \beta V(u(X))^{-1}V(e_j)],
\]

which implies that

\[
\log \det V((1 - \beta)u(X) + \beta e_j) = \log \det V(u(X)) + d\log(1 - \beta) + \log(1 + \beta(\rho - 1))
\]

5
for any \( \beta \in (0,1) \). We now set \( \beta^* = \gamma'/(\rho - 1) \) into the equation above. Note that \( \beta^* \in (0,1) \) since \( \rho > (d+1)(1 + \gamma') \) by (16).

\[
\log \det V((1 - \beta^*)u(X) + \beta^* e_j) \geq \log \det V(u(X)) - d\log \frac{1}{1 - \beta^*} + \log(1 + \gamma') \geq \log \det V(u(X)) - d\log \left(1 + \frac{\gamma'}{\rho - 1 - \gamma'}\right) + \log(1 + \gamma') \geq \log \det V(u(X)) - \frac{\gamma'}{1 + \gamma'} + \log(1 + \gamma'),
\]

where we used \( \log(1 + x) \leq x \) for all \( x > -1 \) to derive the third inequality and \( \rho - 1 - \gamma' > d(1 + \gamma') \) to derive the last inequality.

**Corollary 3.1** Let \( X \subseteq S \). Suppose that (16) holds. Let \( X' \leftarrow X \cup \{p_j\} \). Then

\[
\log \text{vol MVEE}(X') \geq \log \text{vol MVEE}(X) + C_1 \quad \text{if} \quad \gamma \geq \frac{d+1}{d},
\]

\[
\log \text{vol MVEE}(X') \geq \log \text{vol MVEE}(X) + C_2 \epsilon^2 \quad \text{if} \quad 0 < \gamma < \frac{d+1}{d},
\]

where \( C_1 \) and \( C_2 \) are positive constants.

**Proof.** Note that Lemma 3.2 yields a feasible solution \( \tilde{u} \in \mathbb{R}^n \) for (D(S)) with the property that \( \tilde{u}_j = 0 \) for all \( j \notin X' \). Therefore, the restriction of \( \tilde{u} \) to those components corresponding to points in \( X' \) is feasible for the maximization problem (D(X')). Let \( u_{X'} \) denote an optimal solution for (D(X')). We have \( \log \det V(u_{X'}) \geq \log \det V(\tilde{u}) \). Combining this inequality with the result of Lemma 3.2 and using (12), we obtain

\[
\log \text{vol MVEE}(X') \geq \log \text{vol MVEE}(X) + \frac{1}{2} \left( \log(1 + \gamma') - \frac{\gamma'}{1 + \gamma'} \right),
\]

where \( \gamma' \) is defined as in (18). It follows from (15) that \( \gamma' \geq 1 \) if \( \gamma \geq \frac{d+1}{d} \), in which case the second term on the right-hand side of (23) is bounded below by \( C_1 = \frac{1}{2} \) since it is a monotone increasing function of \( \gamma' \). If \( \gamma < \frac{d+1}{d} \), then \( \gamma' < 1 \). It follows from the Taylor expansion that \( \frac{1}{2} \gamma'^2 \) is a lower bound for the second term on the right-hand side of (23). By definition of \( \gamma' \), \( \gamma'^2 \geq \gamma^2/4 \geq \epsilon^2 \), where the last inequality follows from (15). Hence, \( C_2 = 1/2 \).

### 3.3 Core Set Size and Analysis of Algorithm 1

We now combine our results from the previous subsections to derive our main core set result.

**Theorem 3.1** Let \( \gamma \) be as defined in (15). There exists a core set \( X \subseteq S \) such that

\[
|X| = \begin{cases} 
O\left(\frac{d\log d}{\epsilon^2}\right) & \text{if } \gamma \in (0, \frac{d+1}{d}), \\
O(d \log d) & \text{if } \gamma \geq \frac{d+1}{d}.
\end{cases}
\]

**Proof.** We start with the initial core set result of Lemma 3.1. It follows that

\[
\log \text{vol MVEE}(X_0) \leq \log \text{vol MVEE}(S) \leq 2d \log d + \log \text{vol MVEE}(X_0).
\]

By Corollary 3.1, each core set update leads to at least a constant additive increase in \( \log \text{vol MVEE} \) if \( \gamma \in (0, \frac{d+1}{d}) \) and at least an increase of \( \epsilon^2/2 \) if \( \gamma > \frac{d+1}{d} \). At termination, the procedure returns a subset \( X \subseteq S \) such that \( (1 + \epsilon)\text{MVEE}(X) \supseteq S \). However, \( \text{vol MVEE}(X) \leq \text{vol MVEE}(S) \leq \text{vol (1 + \epsilon)MVEE}(X) \). Therefore, \( X \) is a core set of \( S \).
The next theorem establishes a smaller core set size by appropriately decreasing $\epsilon$ to the desired accuracy.

**Theorem 3.2** There exists a core set $X \subseteq S$ with $|X| = \mathcal{O}(d \log d + \frac{d}{\epsilon})$ such that $(1 + \epsilon)\text{MVEE}(X)$ is a $(1 + \epsilon)$-approximation to $\text{MVEE}(S)$.

**Proof.** If $\epsilon > 0$ is chosen large enough such that $\gamma > \frac{d+1}{d}$ (cf. (15)), then the result follows from Theorem 3.1 since, in this case, $\epsilon = \Omega(1)$. Suppose that a smaller value for $\epsilon$ is chosen. Without loss of generality, assume that $\epsilon = e^{-m}$, where $e$ is the base of the natural logarithm. First we apply Theorem 3.1 with $\epsilon_1 = e^{-1}$. This returns a core set $X_1$ of size $\mathcal{O}(d \log d)$. We will now set $\epsilon_i = e^{-i}$, $i = 1, \ldots, m$. We will call these rounds. For $i > 1$, round $i$ will start with the core set $X_{i-1}$ of the previous round, add points to it and will return $X_i$. We now analyze how many points are added to $X_i$ when we change $\epsilon$ from $\epsilon_i = e^{-i}$ to $\epsilon_{i+1} = e^{-(i+1)}$ in the $(i+1)$th round. Let $V_F = \text{vol MVEE}(S)$ and $V_i = \text{vol MVEE}(X_i)$, $i = 1, \ldots, m$. Since $X_i \subseteq S$ is a core set for $\epsilon_i$, we have

$$(1 + \epsilon_i)^d V_i \geq V_F \geq V_i \left(e^{C \epsilon_i^2}ight)^{k_{i+1}},$$

where $C = \frac{1}{2}$, which implies

$$(1 + \epsilon_i)^d \geq \left(e^{C \epsilon_i^2}ight)^{k_{i+1}}.$$ 

Taking logarithms on each side yields

$$d \log (1 + \epsilon_i) \geq C \epsilon_i^2 k_{i+1} \geq \frac{C k_{i+1}}{e^{2i+2}},$$

which implies

$$k_{i+1} \leq \frac{1}{C} e^{2i+2} d \log \left(1 + \epsilon_i\right) \leq \frac{d e^{2i+2}}{C e^i} = \frac{1}{C} d e^{i+2},$$

where we used $\log(1 + x) \leq x$ to derive the second inequality. It follows then that the size of the core set $X$ at the end of round $m$ is given by

$$|X| = |X_1| + \sum_{i=2}^{m} k_i = |X_1| + \frac{1}{C} d \sum_{i=2}^{m} e^{i+1} = \mathcal{O}(d \log d) + \mathcal{O}(d e^m) = \mathcal{O}(d \log d + \frac{d}{\epsilon}),$$

since $\epsilon = e^{-m}$. \hfill \qed

**Theorem 3.3** Algorithm 1 computes a $(1 + \epsilon)$-approximation to the minimum volume enclosing ellipsoid of a given point set $S = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^d$ in at most

$$\mathcal{O}(nd^2 \alpha + \alpha^{4.5} \log \frac{\alpha}{\epsilon})$$

operations, where

$$\alpha := \mathcal{O}(d \log d + \frac{d}{\epsilon})$$

denotes the size of the core set $X \subseteq S$.

**Proof.** We use Khachiyan’s algorithm [17] to compute $\text{MVEE}(X)$ in each round of Algorithm 1, where $X$ denotes the current core set. The initial core set $X_0$ can be computed in $\mathcal{O}(nd^2)$ operations by Lemma 3.1. At each iteration, Algorithm 1 computes $\text{MVEE}(X)$ of the current core set $X$ using Khachiyan’s algorithm, which takes at most $\mathcal{O}(\alpha^{3.5} \log \frac{d}{\epsilon})$ operations since $|X| \leq \alpha$. The stopping condition requires scanning the point set and identifying a maximum violator, both of which can be performed in $\mathcal{O}(nd^2)$ operations. By Theorem 3.2, Algorithm 1 terminates after at most $\alpha$ iterations. Therefore, the total running time of Algorithm 1 is given by (26). \hfill \qed
4 Discussion

In this paper, we developed a $(1+\epsilon)$-approximation algorithm to compute the minimum volume enclosing ellipsoid (MVEE) of a point set $S = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^d$. Our algorithm uses a column generation approach which is especially suited for instances of the MVEE problem with $n \gg d$. We establish the existence of a core set $X \subseteq S$ of size $O(d\log d + \frac{d}{\epsilon^2})$. Using Khachiyan’s algorithm to solve each subproblem yields an approximation algorithm with an improved complexity result for large-scale instances with $n \gg d$ and reasonable values of $\epsilon$.

The approach presented in this paper can be extended to the cases when the set $S$ consists of (full-dimensional) balls or ellipsoids. If such input sets are viewed as a collection of infinitely many points, Lemma 3.1 can easily be extended and yields an initial core set of $2d$ points in $O(nd^2)$ and $O(nd^3)$ operations for balls and ellipsoids, respectively (see Lemma 5.1). The major difference arises in checking stopping conditions and identifying maximum violators. In contrast to the set of points, the problem of whether a given ellipsoid contains another ball or another ellipsoid is in general a nontrivial problem. Either of these problems can be reduced to a semidefinite programming (SDP) feasibility problem involving a $(d+1) \times (d+1)$ matrix [4] (see Lemma 5.2 in the Appendix). Consequently, the stopping condition can still be checked in polynomial time for input sets of balls or ellipsoids. However, if the stopping condition is not satisfied, Algorithm 1 requires the identification of a maximum violator. In contrast to the case of points, finding a maximum violator is a harder problem for the case of balls or ellipsoids. Note that the stopping condition and the maximum violator identification should be performed at every iteration of Algorithm 1 for each of the balls or ellipsoids in the input set. Therefore, Algorithm 1 may not be practical for large number of balls or ellipsoids.

We also note that the MVEE problem can be formulated directly as an SDP problem if the input set consists of balls or ellipsoids [4] and can be solved to an arbitrary precision in polynomial time using interior-point methods [26]. Furthermore, the MVEE problem of a given set of balls can be formulated as an instance of a second-order cone programming problem, which admits more efficient algorithms than SDP, at the expense of increased number of constraints [1].

References


The following lemma extends the result of Lemma 3.1 to input sets consisting of balls and ellipsoids.

**Lemma 5.1** The core set initialization (cf. Lemma 3.1) step can be executed in $O(nd^2)$ and $O(nd^3)$ operations if the input set $S$ consists of $n$ balls or $n$ ellipsoids, respectively.

**Proof.** The core set initialization step involves optimizing linear functions $b_i^T x$, $i = 1, \ldots, d$, over the input set $S \subseteq \mathbb{R}^d$. Assume without loss of generality that $\|b_i\| = 1$, $i = 1, \ldots, d$. For a given ball with center $c \in \mathbb{R}^d$ and radius $r$, the minimum and maximum values of the linear functions $b_i^T x$, $i = 1, \ldots, d$ are achieved at $c - rb_i$ and $c + rb_i$, respectively, which implies that the initialization step can be carried out in $O(nd^2)$ operations if the input set $S$ consists of $n$ balls. For an ellipsoid $E_{Q,c}$, a simple manipulation of optimality conditions reveals that the linear functions $b_i^T x$, $i = 1, \ldots, d$ are minimized and maximized at $c - \frac{1}{\sqrt{b_i^T Q^{-1} b_i}} Q^{-1} b_i$ and $c + \frac{1}{\sqrt{b_i^T Q^{-1} b_i}} Q^{-1} b_i$, respectively. Before the initialization step, we can compute the Cholesky factorizations of the positive definite matrices $Q_j$ for each of the ellipsoids $E_{Q_j,c_j}$, $j = 1, \ldots, n$ in a total of $O(nd^3)$ operations. Therefore, for each ellipsoid, the minimizer and the maximizer can be computed in $O(d^2)$ operations by solving two triangular systems. Consequently, the initialization step requires $O(nd^3)$ operations if $S$ consists of $n$ ellipsoids.

The next lemma shows that stopping condition in Algorithm 1 can be implemented in polynomial time for input sets of balls or ellipsoids.

**Lemma 5.2** The problem of checking whether a given ellipsoid contains a ball or another ellipsoid can be solved in polynomial time.
Proof. We prove the assertion for ellipsoids only since a ball can be viewed as a special type of ellipsoid. It follows from [4] that an ellipsoid $E_{Q,c}$ contains another ellipsoid $E_{Q_j,c_j}$ if and only if there exists a $\tau \geq 0$ such that

$$\tau \begin{bmatrix} Q_j & \quad (-Q_j c_j)^T \\ -Q_j c_j & \quad \sigma_j \end{bmatrix} - \begin{bmatrix} Q & \quad (-Q c)^T \\ -Q c & \quad \sigma \end{bmatrix}$$

(28)

is positive semidefinite, where $\sigma = c^T Q c - 1$ and $\sigma_j = c_j^T Q_j c_j - 1$. However, (28) can be solved in polynomial time both in the bit and the real number models of computation in $\mathcal{O}(d^4)$ operations [27].