An efficient approximation for point-set diameter in higher dimensions

Mahdi Imanparast* Seyed Naser Hashemi* Ali Mohades*†

Abstract

In this paper, we study the problem of computing the diameter of a set of n points in d-dimensional Euclidean space for a fixed dimension d, and propose a new $(1+\varepsilon)$-approximation algorithm with $O(n + 1/\varepsilon^{d-2})$ time and $O(n)$ space, where $0 < \varepsilon \leqslant 1$. We also show that the proposed algorithm can be modified to a $(1 + O(\varepsilon))$-approximation algorithm with $O(n+1/\varepsilon^{\frac{2d}{3}-\frac{1}{2}})$ running time. These results provide some improvements in comparison with existing algorithms in terms of simplicity, and data structure.

1 Introduction

Given a finite set $\mathcal{S}$ of n points, the diameter of $\mathcal{S}$, denoted by $D(\mathcal{S})$ is the maximum distance between two points of $\mathcal{S}$. Namely, we want to find a diametrical pair p and q such that $D(\mathcal{S}) = \max_{p,q\in\mathcal{S}}(||p-q||)$. Computing the diameter of a set of points has a large history, and it may be required in various fields such as database, data mining, and vision. A trivial brute-force algorithm for this problem takes $O(dn^2)$ time, but this is too slow for large-scale data sets that occur in the fields. Hence, we need a faster algorithm which may be exact or is an approximation.

By reducing from the set disjointness problem, it can be shown that computing the diameter of n points in $\mathbb{R}^d$ requires $\Omega(n \log n)$ operations in the algebraic computation-tree model [1]. It is shown by Yao that it is possible to compute the diameter in sub-quadratic time in each dimension [2]. There are well-known solutions in two and three dimensions. In the plane, this problem can be computed in optimal time $O(n \log n)$, but in three dimensions, it is more difficult. Clarkson and Shor [3] present an $O(n \log n)$-time randomized algorithm. Their algorithm needs to compute the intersection of n balls (with the same radius) in $\mathbb{R}^3$. It may be slower than the brute-force algorithm for the most practical data sets, and it is not an efficient method for higher dimensions because the intersection of n balls with the same radius has a large size. Some deterministic algorithms with running time $O(n \log^3 n)$ and $O(n \log^2 n)$ are found for this problem in three dimensions. Finally, Ramos [4] introduced an optimal deterministic $O(n \log n)$-time algorithm in $\mathbb{R}^3$.

In the absence of fast algorithms, many attempts have been made to approximate the diameter in low and high dimensions. A 2-approximation algorithm in $O(dn)$ time can be found easily by selecting a point of $\mathcal{S}$ and then finding the farthest point of it by brute-force manner for the dimension d. The first non-trivial approximation algorithm for the diameter is presented by Egecioglu and Kalantari [5] that approximates the diameter with factor $\sqrt{3}$ and operations cost $O(dn)$. They also present an iterative algorithm with $t \leq n$ iterations and the cost $O(dn)$ for each iteration that has approximate factor $\sqrt{5-2\sqrt{3}}$. Agarwal et al. [6] present a $(1+\varepsilon)$-approximation algorithm in $\mathbb{R}^d$ with $O(n/\varepsilon^{(d-1)/2})$ running time by projection to directions. Barequet and Har Peled [7] present a $\sqrt{d}$-approximation diameter method with $O(dn)$ time. They also describe a $(1 + \varepsilon)$-approximation approach with $O(n + 1/\varepsilon^{2d})$ time. They show that the running time can be improved to $O(n + 1/\varepsilon^{2(d-1)})$. Similarly, Har Peled [8] presents an approach which for the most inputs is able to compute very fast the exact diameter, or an approximation with $O((n + 1/\varepsilon^{2d}) \log 1/\varepsilon)$ running time. Although, in the worst case, the algorithm running time is still quadratic, and it is sensitive to the hardness of the input. Chan [9] observes that a combination of two approaches in [6] and [7] yields a $(1 + \varepsilon)$-approximation with $O(n + 1/\varepsilon^{3(d-1)/2})$ time and a $(1 + O(\varepsilon))$-approximation with $O(n + 1/\varepsilon^{d-\frac{1}{2}})$ time. He also introduces a core-set theorem, and shows that using this theorem, a $(1 + O(\varepsilon))$-approximation in $O(n + 1/\varepsilon^{d-\frac{3}{2}})$ time can be found [10]. Recently, Chan [11] has proposed an approximation algorithm with $O((n/\sqrt{\varepsilon} + 1/\varepsilon^{\frac{d}{2}+1})(\log \frac{1}{\varepsilon})^{O(1)})$ time by applying the Chebyshev polynomials in low constant dimensions, and Arya et al. [12] show that by applying an efficient decomposition of a convex body using a hierarchy of Macbeath regions, it is possible to compute an approximation in $O(n \log \frac{1}{\varepsilon} + 1/\varepsilon^{\frac{(d-1)}{2}+\alpha})$ time, where α is an arbitrarily small positive constant.

1.1 Our results

In this paper, we propose a new $(1 + \varepsilon)$-approximation algorithm for computing the diameter of a set $\mathcal{S}$ of

*Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran, m.imanparast@aut.ac.ir, nhashemi@aut.ac.ir

†Laboratory of Algorithms and Computational Geometry, Amirkabir University of Technology, Tehran, Iran, mohades@aut.ac.ir

Table 1: A summary on the complexity of some non-constant approximation algorithm for the diameter of a point set. Our results are denoted by +.

Ref.	Approx. Factor	Running Time
[6]	$1+\varepsilon$	$O(\frac{n}{\varepsilon^{(d-1)/2}})$
[7]	$1+\varepsilon$	$O(n+1/\varepsilon^{2(d-1)})$
[8]	$1+\varepsilon$	$O((n+1/\varepsilon^{2d})\log\frac{1}{\varepsilon})$
[9]	$1+\varepsilon$	$O(n+1/\varepsilon^{\frac{3(d-1)}{2}})$
+	$1+\varepsilon$	$O(n+1/\varepsilon^{d-2})$
[9]	$1+O(\varepsilon)$	$O(n+1/\varepsilon^{d-\frac{1}{2}})$
[10]	$1+O(\varepsilon)$	$O(n+1/\varepsilon^{d-\frac{3}{2}})$
[11]	$1+O(\varepsilon)$	$O((\frac{n}{\sqrt{\varepsilon}}+1/\varepsilon^{\frac{d}{2}+1})(\log\frac{1}{\varepsilon})^{O(1)})$
[12]	$1+O(\varepsilon)$	$O(n\log\frac{1}{\varepsilon}+1/\varepsilon^{\frac{(d-1)}{2}+\alpha})$
+	$1+O(\varepsilon)$	$O(n+1/\varepsilon^{\frac{2d}{3}-\frac{1}{2}})$

n points in $\mathbb{R}^d$ with $O(n+1/\varepsilon^{d-2})$ time and $O(n)$ space, where $0<\varepsilon\leqslant 1$. Moreover, we show that the proposed algorithm can be modified to a $(1+O(\varepsilon))$-approximation algorithm with $O(n+1/\varepsilon^{\frac{2d}{3}-\frac{1}{2}})$ time and $O(n)$ space. As stated above, two new results have been recently presented for this problem in [11] and [12]. It should be noted that our algorithms are completely different in terms of computational technique. The polynomial technique provided by Chan [11] is based on using Chebyshev polynomials and discrete upper envelope subroutine [10], and the method presented by Arya et al. [12] requires the use of complex data structures to approximately answer queries for polytope membership, directional width, and nearest-neighbor. While our algorithms in comparison with these algorithms are simpler in terms of understanding and data structure. We have provided a summary on the non-constant approximation algorithms for the diameter in Table 1.

2 The proposed algorithm

In this section, we describe our new approximation algorithm to compute the diameter of a point set. In our algorithm, we first find the extreme points in each coordinate and compute the axis-parallel bounding box of $\mathcal{S}$, which is denoted by $B(\mathcal{S})$. We use the largest length side ℓ of $B(\mathcal{S})$ to impose grids on the point set. In fact, we first decompose $B(\mathcal{S})$ to a grid of regular hypercubes with side length ξ, where $\xi=\varepsilon\ell/2\sqrt{d}$. We call each hypercube a cell. Then, each point of $\mathcal{S}$ is rounded to its corresponding central cell-point. See Figure 1. In the following, we impose again an ξ_1-grid to $B(\mathcal{S})$ for $\xi_1=\sqrt{\varepsilon}\ell/2\sqrt{d}$. We round each point of the rounded point set $\hat{\mathcal{S}}$ to its nearest grid-point in this new grid that

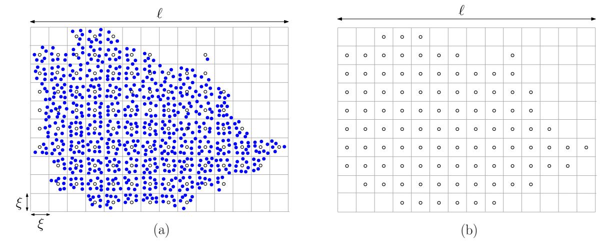

Figure 1: (a) A set of points in $\mathbb{R}^2$ and an ξ-grid. Initial points are shown by blue points and their corresponding central cell-points are shown by circle points. (b) Rounded point set $\hat{\mathcal{S}}$.

results in a point set $\hat{\mathcal{S}}_1$. Let, $\mathcal{B}_\delta(p)$ be a hypercube with side length δ and central-point p. We restrict our search domain for finding diametrical pairs of the first rounded point set $\hat{\mathcal{S}}$ into two hypercubes $\mathcal{B}_{2\xi_1}(\hat{p}_1)$ and $\mathcal{B}_{2\xi_1}(\hat{q}_1)$ corresponding to two diametrical pair points $\hat{p}_1$ and $\hat{q}_1$ in the point set $\hat{\mathcal{S}}_1$. Let us use two point sets $\mathcal{B}_1$ and $\mathcal{B}_2$ for maintaining points of the rounded point set $\hat{\mathcal{S}}$, which are inside two hypercubes $\mathcal{B}_{2\xi_1}(\hat{p}_1)$ and $\mathcal{B}_{2\xi_1}(\hat{q}_1)$, respectively (see Figure 2). Then, it is sufficient to find a diameter between points of $\hat{\mathcal{S}}$, which are inside two point sets $\mathcal{B}_1$ and $\mathcal{B}_2$. We use notation $Diam(\mathcal{B}_1,\mathcal{B}_2)$ for the process of computing the diameter of the point set $\mathcal{B}_1\cup\mathcal{B}_2$. Altogether, we can present the following algorithm.

Algorithm 1: APPROXIMATE DIAMETER $(\mathcal{S},\varepsilon)$

Input: a set $\mathcal{S}$ of n points in $\mathbb{R}^d$ and an error parameter ε.

Output: Approximate diameter $\tilde{D}$.

Compute the axis-parallel bounding box $B(\mathcal{S})$ for the point set $\mathcal{S}$.
$\ell \leftarrow$ Find the length of the largest side in $B(\mathcal{S})$.
Set $\xi \leftarrow \varepsilon\ell/2\sqrt{d}$ and $\xi_1 \leftarrow \sqrt{\varepsilon}\ell/2\sqrt{d}$.
$\hat{\mathcal{S}} \leftarrow$ Round each point of $\mathcal{S}$ to its central-cell point in a ξ-grid.
$\hat{\mathcal{S}}_1 \leftarrow$ Round each point of $\hat{\mathcal{S}}$ to its nearest grid-point in a ξ_1-grid.
$\hat{D}_1 \leftarrow$ Compute the diameter of the point set $\hat{\mathcal{S}}_1$ by brute-force manner, and simultaneously, a list of the diametrical pairs $(\hat{p}_1,\hat{q}_1)$, such that $\hat{D}_1=||\hat{p}_1-\hat{q}_1||$.
Find points of $\hat{\mathcal{S}}$ which are in two hypercubes $\mathcal{B}_1=\mathcal{B}_{2\xi_1}(\hat{p}_1)$ and $\mathcal{B}_2=\mathcal{B}_{2\xi_1}(\hat{q}_1)$, for each diametrical pair $(\hat{p}_1,\hat{q}_1)$.
$\hat{D} \leftarrow$ Compute $Diam(\mathcal{B}_1,\mathcal{B}_2)$, corresponding to each diametrical pair $(\hat{p}_1,\hat{q}_1)$ by brute-force manner and return the maximum value between them.
$\tilde{D} \leftarrow \hat{D}+\varepsilon\ell/2$.
Output $\tilde{D}$.

2.1 Analysis

In this subsection, we analyze the proposed algorithm.

Theorem 1 *Algorithm 1 computes an approximate diameter for a set $\mathcal{S}$ of n points in $\mathbb{R}^d$ in $O(n+1/\varepsilon^{d-2})$ time and $O(n)$ space, where $0<\varepsilon\leqslant 1$.*

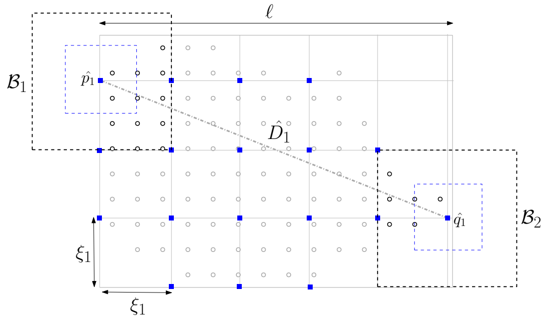

Figure 2: Points of the set $\hat{\mathcal{S}}$ are shown by circle points and their corresponding nearest grid-points in set $\hat{\mathcal{S}}_1$ are shown by blue square points.

Proof. Finding the extreme points in all coordinates and finding the largest side of $B(\mathcal{S})$ can be done in $O(dn)$ time. The rounding step takes $O(d)$ time for each point, and for all of them takes $O(dn)$ time. But for computing the diameter over the rounded point set $\hat{\mathcal{S}}_1$ we need to know the number of points in the set $\hat{\mathcal{S}}_1$. We know that the largest side of the bounding box $B(\mathcal{S})$ has length ℓ and the side length of each cell in ξ_1-grid is $\xi_1 = \sqrt{\varepsilon}\ell/2\sqrt{d}$. On the other hand, the volume of a hypercube of side length L in d-dimensional space is L^d. Since, corresponding to each point in the point set $\hat{\mathcal{S}}_1$, we can take a hypercube of side length ξ_1. Therefore, in order to count the maximum number of points inside the set $\hat{\mathcal{S}}_1$, it is sufficient to calculate the number of hypercubes of length ξ_1 in a hypercube (bounding box) with length $\ell + \xi_1$. See Figure 2. This means that the number of grid-points in an imposed ξ_1-grid to the bounding box $B(\mathcal{S})$ is at most

$$\frac{(\ell+\xi_1)^d}{(\xi_1)^d} = \left(\frac{2\sqrt{d}}{\sqrt{\varepsilon}}+1\right)^d = O\left(\frac{(2\sqrt{d})^d}{\varepsilon^{\frac{d}{2}}}\right). \quad (1)$$

So, the number of points in $\hat{\mathcal{S}}_1$ is at most $O((2\sqrt{d})^d/\varepsilon^{\frac{d}{2}})$. Hence, by the brute-force quadratic algorithm, we need $O(((2\sqrt{d})^d/\varepsilon^{\frac{d}{2}})^2) = O((2\sqrt{d})^{2d}/\varepsilon^d)$ time for computing all distances between grid-points of the set $\hat{\mathcal{S}}_1$, and its diametrical pair list. Then, for a diametrical pair $(\hat{p}_1, \hat{q}_1)$ in the point set $\hat{\mathcal{S}}_1$, we compute two sets $\mathcal{B}_1$ and $\mathcal{B}_2$. This work takes $O(dn)$ time. In addition, for computing the diameter of point set $\mathcal{B}_1 \cup \mathcal{B}_2$, we need to know the number of points in each of them. On the other hand, the number of points in two sets $\mathcal{B}_1$ or $\mathcal{B}_2$ is at most

$$\frac{Vol(\mathcal{B}_{2\xi_1})}{Vol(\mathcal{B}_\xi)} = \frac{(2\sqrt{\varepsilon}\ell/2\sqrt{d})^d}{(\varepsilon\ell/2\sqrt{d})^d} = \frac{(2\sqrt{\varepsilon})^d}{\varepsilon^d} = \frac{(2)^d}{\varepsilon^{\frac{d}{2}}}. \quad (2)$$

Hence, for computing $Diam(\mathcal{B}_1, \mathcal{B}_2)$, we need $O(((2)^d/\varepsilon^{\frac{d}{2}})^2) = O((2)^{2d}/\varepsilon^d)$ time by brute-force manner, but we might have more than one diametrical pair $(\mathcal{B}_1, \mathcal{B}_2)$. Since the point set $\hat{\mathcal{S}}_1$ is a set of grid-points, so we could have in the worst-case $O(2^d)$ different diametrical pairs $(\mathcal{B}_1, \mathcal{B}_2)$ in the point set $\hat{\mathcal{S}}_1$. This means that this step takes at most $O(2^d \cdot (2)^{2d}/\varepsilon^d) = O((2\sqrt{2})^{2d}/\varepsilon^d)$ time. Now, we can present the complexity of our algorithm as follows:

$$T_d(n) = O(dn) + O\left(\frac{(2\sqrt{d})^{2d}}{\varepsilon^d}\right) + O(2^d dn) + O\left(\frac{(2\sqrt{2})^{2d}}{\varepsilon^d}\right),$$

$$\leqslant O\left(2^d dn + \frac{(2\sqrt{d})^{2d}}{\varepsilon^d}\right). \quad (3)$$

Since d is fixed, we have: $T_d(n) = O(n + \frac{1}{\varepsilon^d})$.

We can also reduce the running time of the Algorithm 1 by discarding some internal points which do not have any potential to be the diametrical pairs in rounded point set $\hat{\mathcal{S}}_1$, and similarly, in two point sets $\mathcal{B}_1$ and $\mathcal{B}_2$. By considering all the points which are same in their $(d-1)$ coordinates and keep only highest and lowest [7]. Then, the number of points in $\hat{\mathcal{S}}_1$, and two point sets $\mathcal{B}_1$ and $\mathcal{B}_2$ can be reduced to $O(1/\varepsilon^{\frac{d}{2}-1})$. So, using the brute-force quadratic algorithm, we need $O((1/\varepsilon^{\frac{d}{2}-1})^2)$ time to find the diametrical pairs. Hence, this gives us the total running time $O(n + 1/\varepsilon^{d-2})$. About the required space, we only need $O(n)$ space for storing required point sets. So, this completes the proof. □

Now, we explain the details of the approximation factor.

Theorem 2 *Algorithm 1 computes an approximate diameter $\tilde{D}$ such that: $D \leqslant \tilde{D} \leqslant (1+\varepsilon)D$, where $0 < \varepsilon \leqslant 1$.*

Proof. In line 7 of the Algorithm 1, we compute two point sets $\mathcal{B}_1$ and $\mathcal{B}_2$, for each diametrical pair $(\hat{p}_1, \hat{q}_1)$ in the point set $\hat{\mathcal{S}}_1$. We know that a grid-point $\hat{p}_1$ in point set $\hat{\mathcal{S}}_1$ is formed from points of the set $\hat{\mathcal{S}}$ which are inside hypercube $B_{\xi_1}(\hat{p}_1)$. We use a hypercube $\mathcal{B}_1$ of side length $2\xi_1$ to make sure that we do not lose any candidate diametrical pair of the first rounded point set $\hat{\mathcal{S}}$ around a diametrical point $\hat{p}_1$ (see Figure 2). In the next step, we should find the diametrical pair $(\hat{p}, \hat{q}) \in \hat{\mathcal{S}}$ for points which are inside two point sets $\mathcal{B}_1$ and $\mathcal{B}_2$. Hence, it is remained to show that the diameter, which is computed by two points $\hat{p}$ and $\hat{q}$, is a $(1+\varepsilon)$-approximation of the true diameter. Let $\hat{p}$ and $\hat{q}$ are two central-cell points of the first rounded point set $\hat{\mathcal{S}}$ which are used in line 8 of the Algorithm 1 for computing the approximate diameter $\hat{D}$. Then, we have two cases, either two true points p and q are in far distance from each other in their corresponding cells (Figure 3 (a)), or they are in near distance from each other (Figure 3 (b)). It is obvious that the other cases are between these two cases.

For first case (Figure 3 (a)), let for two projected points $\hat{p}'$ and $\hat{q}'$, we set $d_1 = ||p - \hat{p}'||$ and $d_2 = ||q - \hat{q}'||$.

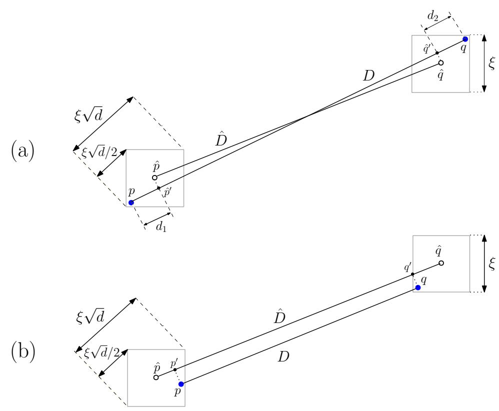

Figure 3: Two cases in proof of the Theorem 2.

We know that the side length of each cell in a grid which is used for $\hat{\mathcal{S}}$ is ξ. So, the hypercube (cell) diagonal is $\xi\sqrt{d}$. From Figure 3 (a) it can be found that $d_1 < \xi\sqrt{d}/2$ and $d_2 < \xi\sqrt{d}/2$. Therefore, we have

$$\begin{aligned} D &= \hat{D} + d_1 + d_2, \\ D &\leqslant \hat{D} + \xi\sqrt{d}, \\ D - \xi\sqrt{d} &\leqslant \hat{D}. \end{aligned} \tag{4}$$

Similarly, for the second case (Figure 3 (b)), we know that $c_1 = ||\hat{p} - p'|| < \xi\sqrt{d}/2$ and $c_2 = ||\hat{q} - q'|| < \xi\sqrt{d}/2$. So,

$$\begin{aligned} \hat{D} &= D + c_1 + c_2, \\ \hat{D} &\leqslant D + \xi\sqrt{d}. \end{aligned} \tag{5}$$

Then, from (4) and (5) we can result:

$$D - \xi\sqrt{d} \leqslant \hat{D} \leqslant D + \xi\sqrt{d}. \tag{6}$$

Since we know that $\xi = \varepsilon\ell/2\sqrt{d}$, we have:

$$\begin{aligned} D - \varepsilon\ell/2 \leqslant \hat{D} \leqslant D + \varepsilon\ell/2, \\ D \leqslant \hat{D} + \varepsilon\ell/2 \leqslant D + \varepsilon\ell. \end{aligned} \tag{7}$$

We know that $\ell \leqslant D$. For this reason we can result:

$$D \leqslant \hat{D} + \varepsilon\ell/2 \leqslant (1+\varepsilon)D. \tag{8}$$

Finally, if we assume that $\tilde{D} = \hat{D} + \varepsilon\ell/2$, we have:

$$D \leqslant \tilde{D} \leqslant (1+\varepsilon)D. \tag{9}$$

Therefore, the theorem is proven. □

2.2 The modified algorithm

In this subsection, we present a modified version of our proposed algorithm by combining it with a recursive approach due to Chan [9]. Hence, we first explain Chan's recursive approach. As mentioned before, Agarwal et al. [6] proposed a $(1+\varepsilon)$-approximation algorithm for computing the diameter of a set of points in $\mathbb{R}^d$. Their result is based on the following simple fact that we can find $O(1/\varepsilon^{(d-1)/2})$ numbers of directions in $\mathbb{R}^d$, for example by constructing a uniform grid on a unit sphere, such that for each vector $x \in \mathbb{R}^d$, there is a direction that the angle between this direction and x be at most $\sqrt{\varepsilon}$. In fact, they found a small set of directions which can approximate well all directions. This can be done by forming unit vectors which start from origin to grid-points of a uniform grid on a unit sphere [6], or to grid-points on the boundary of a box [10]. These sets of directions have cardinality $O(1/\varepsilon^{(d-1)/2})$. The following observation explains how we can find these directions on the boundary of a box.

Observation 1 *([10]) Consider a box B which includes origin o such that the boundary of this box (∂B) be in the distance at least 1 from the origin. For a $\sqrt{\varepsilon/2}$-grid on ∂B and for each vector $\vec{x}$, there is a grid point $\boldsymbol{a}$ on ∂B such that the angle between two vectors $\vec{a}$ and $\vec{x}$ is at most $arccos(1-\varepsilon/8) \leqslant \sqrt{\varepsilon}$.*

This observation explains that grid-points on the boundary of a box (∂B) form a set V_d of $O(1/\varepsilon^{(d-1)/2})$ numbers of unit vectors in $\mathbb{R}^d$ such that for each $x \in \mathbb{R}^d$, there is a vector $a \in V_d$ from the origin o to a grid-point a on ∂B, where the angle between two vectors x and a is at most $\sqrt{\varepsilon}$. On the other hand, according to observation 1, there is a vector $a \in V_d$ such that if α be the angle between two vectors x and a, then, $\alpha \leqslant arccos(1-\varepsilon/8)$, and so $cos\alpha \geqslant (1-\varepsilon/8)$. If x' is the projection of the vector x on the vector a, then:

$$\begin{aligned} ||x|| = \frac{||x'||}{cos\alpha} &\leqslant ||x'|| \frac{1}{(1-\frac{\varepsilon}{8})} \\ &\leqslant ||x'||(1 + \frac{\varepsilon}{8} + \frac{\varepsilon^2}{8^2} + \frac{\varepsilon^3}{8^3} + \cdots) \\ &\leqslant ||x'||(1+\varepsilon). \end{aligned} \tag{10}$$

So, we have:

$$||x'|| \leqslant ||x|| \leqslant (1+\varepsilon)||x'||. \tag{11}$$

This means that if pair (p,q) be the diametrical pair of a point set, then there is a vector $a \in V_d$ such that the angle between two vectors pq and a is at most $\sqrt{\varepsilon}$. See Figure 4. Then, pair (p',q') which is the projection of the pair (p,q) on the vector a, is a $(1+\varepsilon)$-approximation of the true diametrical pair (p,q), and we have:

$$||p'-q'|| \leqslant ||p-q|| \leqslant (1+\varepsilon)||p'-q'||. \tag{12}$$

In other words, we can project point set $\mathcal{S}$ on $O(1/\varepsilon^{(d-1)/2})$ directions for all $a \in V_d$, and compute a $(1+\varepsilon)$-approximation of the diameter by finding maximum diameter between all directions. We project n

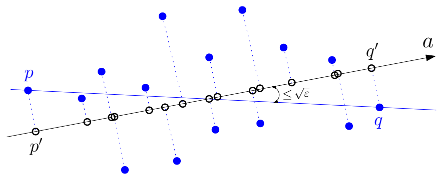

Figure 4: Projecting a point set on a direction a.

points on $|V_d| = O(1/\varepsilon^{(d-1)/2})$ directions. Since, computing the extreme points on each direction $a \in V_d$ takes $O(n)$ time. Consequently, Agarwal et al. [6] algorithm computes a $(1+\varepsilon)$-approximation of the diameter in $O(n/\varepsilon^{(d-1)/2})$ time. Chan [9] proposes that if we reduce the number of points from n to $O(1/\varepsilon^{d-1})$ by rounding to a grid and then apply Agarwal et al. [6] method on this rounded point set, we need $O((1/\varepsilon^{d-1})/\varepsilon^{(d-1)/2}) = O(1/\varepsilon^{3(d-1)/2})$ time to compute the maximum diameter over all $O(1/\varepsilon^{(d-1)/2})$ directions. Taking into account $O(n)$ time for rounding to a grid, this new approach takes $O(n + 1/\varepsilon^{3(d-1)/2})$ time. Moreover, Chan [9] observed that the bottleneck of this approach is the large number of projection operations. Hence, he proposes that we can project points on a set of $O(1/\sqrt{\varepsilon})$ 2-dimensional unit vectors instead of $O(1/\varepsilon^{(d-1)/2})$ d-dimensional unit vectors to reduce the problem to $O(1/\sqrt{\varepsilon})$ numbers of $(d-1)$-dimensional subproblems which can be solved recursively. In fact, according to the relation (11), for a vector $x \in \mathbb{R}^2$, there is a vector a such that:

$$||x'|| \leqslant ||x|| \leqslant (1+\varepsilon)||x'||, \quad x \in \mathbb{R}^2. \qquad (13)$$

where x' is the projection of the vector x on vector a. Since a is a unit vector ($||a|| = 1$), therefore, $||x'|| = (a \cdot x)/||a|| = a \cdot x$. Hence, we can rewrite the previous relation as follows:

$$(a \cdot x)^2 \leqslant ||x||^2 \leqslant (1+\varepsilon)^2 (a \cdot x)^2, \ x \in \mathbb{R}^2, a \in V_2, \quad (14)$$

or

$$(a_1x_1 + a_2x_2)^2 \leqslant (x_1^2 + x_2^2) \leqslant (1+\varepsilon)^2 (a_1x_1 + a_2x_2)^2, \ a \in V_2. \qquad (15)$$

where x_i be the ith coordinate for a point $x \in \mathbb{R}^d$. We can expand (15) to:

$$(a_1x_1 + a_2x_2)^2 + \cdots + x_d^2 \leqslant (x_1^2 + x_2^2 + \cdots + x_d^2) \leqslant (1+\varepsilon)^2((a_1x_1 + a_2x_2)^2 + \cdots + x_d^2). \qquad (16)$$

Now, define the projection $\pi_a : \mathbb{R}^d \to \mathbb{R}^{d-1} : \pi_a(x) = (a_1x_1 + a_2x_2, x_3, \cdots, x_d) \in \mathbb{R}^{d-1}$. Then, we can rewrite relation (16) for each vector $x \in \mathbb{R}^d$ as follows:

$$||\pi_a(x)||^2 \leqslant ||x||^2 \leqslant (1+\varepsilon)^2 ||\pi_a(x)||^2, \quad a \in V_2. \qquad (17)$$

So, since $||\pi_a(p-q)|| = ||\pi_a(p)|| - ||\pi_a(q)||$ we have for diametrical pair (p, q):

$$||\pi_a(p-q)|| \leqslant ||p-q|| \leqslant (1+\varepsilon)||\pi_a(p-q)||, \quad a \in V_2. \qquad (18)$$

Therefore, for finding a $(1+O(\varepsilon))$-approximation for the diameter of point set $P \subseteq \mathbb{R}^d$, it is sufficient that we approximate recursively the diameter of a projected point set $\pi_a(P) \subset \mathbb{R}^{d-1}$ over each of the vectors $a \in V_2$. Then, the maximum diametrical pair computed in the recursive calls is a $(1+O(\varepsilon))$-approximation to the diametrical pair. Now, let us reduce the number of points from n to $m = O(1/\varepsilon^{d-1})$ by rounding to a grid, and we denote the required time for computing the diameter of m points in d-dimensional space with $t_d(m)$. Then, for $m = O(1/\varepsilon^{d-1})$ grid points, this approach breaks the problem into $O(1/\sqrt{\varepsilon})$ subproblems in a $(d-1)$ dimension. Hence, we have a recurrence $t_d(m) = O(m + 1/\sqrt{\varepsilon} t_{d-1}(O(1/\varepsilon^{d-1})))$. By assuming $E = 1/\varepsilon$, we can rewrite the recurrence as:

$$t_d(m) = O(m + E^{\frac{1}{2}} t_{d-1}(O(E^{d-1}))). \qquad (19)$$

This can be solved to: $t_d(m) = O(m + E^{d-\frac{1}{2}})$. In this case, $m = O(1/\varepsilon^{d-1})$, so, this recursive takes $O(1/\varepsilon^{d-\frac{1}{2}})$ time. Taking into account $O(n)$ time, we spent for rounding to a grid at the first, Chan's recursive approach computes a $(1+O(\varepsilon))$-approximation for the diameter of a set of n points in $O(n + 1/\varepsilon^{d-\frac{1}{2}})$ time [9].

In the following, we use Chan's recursive approach in a phase of our proposed algorithm.

Algorithm 2: APPROXIMATE DIAMETER 2 $(\mathcal{S}, \varepsilon)$

Input: a set $\mathcal{S}$ of n points in $\mathbb{R}^d$ and an error parameter ε.

Output: Approximate diameter $\tilde{D}$.

1: Compute the axis-parallel bounding box $B(\mathcal{S})$ for the point set $\mathcal{S}$.
2: $\ell \leftarrow$ Find the length of the largest side in $B(\mathcal{S})$.
3: Set $\xi \leftarrow \varepsilon\ell/2\sqrt{d}$ and $\xi_2 \leftarrow \varepsilon^{\frac{1}{3}}\ell/2\sqrt{d}$.
4: $\hat{\mathcal{S}} \leftarrow$ Round each point of $\mathcal{S}$ to its central-cell point in a ξ-grid.
5: $\hat{\mathcal{S}}_1 \leftarrow$ Round each point of $\hat{\mathcal{S}}$ to its nearest grid-point in a ξ_2-grid.
6: $\hat{D}_1 \leftarrow$ Compute the diameter of the point set $\hat{\mathcal{S}}_1$ by brute-force, and simultaneously, a list of the diametrical pairs $(\hat{p}_1, \hat{q}_1)$, such that $\hat{D}_1 = ||\hat{p}_1 - \hat{q}_1||$.
7: Find points of $\hat{\mathcal{S}}$ which are in two hypercubes $\mathcal{B}_1 = \mathcal{B}_{2\xi_2}(\hat{p}_1)$ and $\mathcal{B}_2 = \mathcal{B}_{2\xi_2}(\hat{q}_1)$ for each diametrical pair $(\hat{p}_1, \hat{q}_1)$.
8: $\tilde{D} \leftarrow$ Compute $Diam(\mathcal{B}_1, \mathcal{B}_2)$, corresponding to each diametrical pair $(\hat{p}_1, \hat{q}_1)$ using Chan's [9] recursive approach and return the maximum value $||p' - q'||$ over all of them.
9: Output $\tilde{D}$.

Now, we will analyze the Algorithm 2.

Theorem 3 *A $(1+O(\varepsilon))$-approximation for the diameter of a set of n points in d-dimensional Euclidean space can be computed in $O(n+1/\varepsilon^{\frac{2d}{3}-\frac{1}{2}})$ time and $O(n)$ space, where $0 < \varepsilon \leqslant 1$.*

Proof. As it can be seen, lines 1 to 6 of the Algorithm 2 are the same as the Algorithm 1. In this case, the number of points in rounded points set $\hat{\mathcal{S}}_1$ is at most:

$$\frac{(\ell+\xi_2)^d}{(\xi_2)^d} = \left(\frac{2\sqrt{d}}{\varepsilon^{\frac{1}{3}}} + 1\right)^d = O\left(\frac{(2\sqrt{d})^d}{\varepsilon^{\frac{d}{3}}}\right). \qquad (20)$$

This can be reduced to $O((2\sqrt{d})^d/\varepsilon^{\frac{d}{3}-1})$, by keeping only highest and lowest points which are the same in their $(d-1)$ coordinates. So, for finding all diametrical pairs of the point set $\hat{\mathcal{S}}_1$, we need $O((2\sqrt{d})^d/\varepsilon^{\frac{d}{3}-1})^2) = O((2\sqrt{d})^{2d}/\varepsilon^{\frac{2d}{3}-2})$ time. Moreover, the number of points in two sets $\mathcal{B}_1$ or $\mathcal{B}_2$ is at most

$$\frac{Vol(\mathcal{B}_{2\xi_2})}{Vol(\mathcal{B}_\xi)} = \frac{(2\varepsilon^{\frac{1}{3}}\ell/2\sqrt{d})^d}{(\varepsilon\ell/2\sqrt{d})^d} = \frac{(2\varepsilon^{\frac{1}{3}})^d}{\varepsilon^d} = \frac{(2)^d}{\varepsilon^{\frac{2d}{3}}}. \quad (21)$$

This can be reduced to $O((2)^d/\varepsilon^{\frac{2d}{3}-1})$. Now, for computing $Diam(\mathcal{B}_1, \mathcal{B}_2)$, we use Chan's [9] recursive approach instead of using the quadratic brute-force algorithm on the point set $\mathcal{B}_1 \cup \mathcal{B}_2$. On the other hand, computing the diameter on a set of $O(1/\varepsilon^{\frac{2d}{3}-1})$ points using Chan's recursive approach takes the following recurrence based on the relation (19): $t_d(m) = O(m + 1/\sqrt{\varepsilon}t_{d-1}(O(1/\varepsilon^{\frac{2d}{3}-1})))$. By assuming $E = 1/\varepsilon$, we can rewrite the recurrence as:

$$t_d(m) = O(m + E^{\frac{1}{2}}t_{d-1}(O(E^{\frac{2d}{3}-1}))). \quad (22)$$

This can be solved to: $t_d(m) = O(m + E^{\frac{2d}{3}-\frac{1}{2}})$. In this case, $m = O(E^{\frac{2d}{3}-1})$, so, this recursive takes $O(E^{\frac{2d}{3}-\frac{1}{2}}) = O(1/\varepsilon^{\frac{2d}{3}-\frac{1}{2}})$ time. Moreover, if we have more than one diametrical pair $(\hat{p}_1, \hat{q}_1)$ in point set $\hat{\mathcal{S}}_1$, then this step takes at most $O((2^d)(2)^d/\varepsilon^{\frac{2d}{3}-\frac{1}{2}}) = O(2^{2d}/\varepsilon^{\frac{2d}{3}-\frac{1}{2}})$ time. So, we have total time:

$$T_d(n) = O(dn) + O\left(\frac{(2\sqrt{d})^{2d}}{\varepsilon^{\frac{2d}{3}-2}}\right) + O(2^d dn) + O\left(\frac{2^{2d}}{\varepsilon^{\frac{2d}{3}-\frac{1}{2}}}\right),$$
$$\leqslant O\left(2^d dn + \frac{(2\sqrt{d})^{2d}}{\varepsilon^{\frac{2d}{3}-\frac{1}{2}}}\right). \quad (23)$$

Since d is fixed, we have: $T_d(n) = O(n + \frac{1}{\varepsilon^{\frac{2d}{3}-\frac{1}{2}}})$.

In addition, Chan's recursive approach in line 8 of the Algorithm 2 returns a diametrical pair (p', q') which is a $(1 + O(\varepsilon))$-approximation for the diametrical pair $(\hat{p}, \hat{q}) \in \hat{\mathcal{S}}$. So, according to relation (12), we have:

$$||p' - q'|| \leqslant ||\hat{p} - \hat{q}|| \leqslant (1 + O(\varepsilon))||p' - q'||. \quad (24)$$

Moreover, the diametrical pair $(\hat{p}, \hat{q})$ is an approximation of the true diametrical pair $(p, q) \in \mathcal{S}$, and according to the relation (8), we have:

$$||p - q|| \leqslant ||\hat{p} - \hat{q}|| + \varepsilon\ell/2 \leqslant (1 + \varepsilon)||p - q||. \quad (25)$$

Hence, from (24) and (25) we can result:

$$\begin{aligned} ||p - q|| &\leqslant ||\hat{p} - \hat{q}|| + \varepsilon\ell/2, \\ &\leqslant ||\hat{p} - \hat{q}|| + \varepsilon||\hat{p} - \hat{q}||, \\ &\leqslant (1 + \varepsilon)||\hat{p} - \hat{q}||, \\ &\leqslant (1 + \varepsilon)((1 + O(\varepsilon))||p' - q'||), \\ &\leqslant (1 + O(\varepsilon))||p' - q'||. \quad (26) \end{aligned}$$

So, Algorithm 2 finds a $(1 + O(\varepsilon))$-approximation in $O(n + 1/\varepsilon^{\frac{2d}{3}-\frac{1}{2}})$ time and $O(n)$ space. □

3 Conclusion

We have presented two new non-constant approximation algorithms to compute the diameter of a point set $\mathcal{S}$ of n points in $\mathbb{R}^d$ for a fixed dimension d, which provide some improvements in terms of simplicity, and data structure.

References

[1] F. P. Preparata, and M. I. Shamos. *Computational Geometry: an Introduction*. New York, Springer-Verlag, pages 176–182, 1985.

[2] A. C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11:721–736, 1982.

[3] K. L. Clarkson, and P. W. Shor. Applications of random sampling in computational geometry. *Discrete and Computational Geometry*, 4:387–421, 1989.

[4] E. A. Ramos. An optimal deterministic algorithm for computing the diameter of a three-dimensional point set. *Discrete and Computational Geometry*, 26:245–265, 2001.

[5] O. Egecioglu, and B. Kalantari. Approximating the diameter of a set of points in the Euclidean space. *Information Processing Letters*, 32(4):205–211, 1989.

[6] P. K. Agarwal, J. Matousek, and S. Suri. Farthest neighbors maximum spanning trees and related problems in higher dimensions. *Computational Geometry: Theory and Applications*, 1:189–201, 1992.

[7] G. Barequet, and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *Journal of Algorithms*, 38:91–109, 2001.

[8] S. Har-Peled. A practical approach for computing the diameter of a point set. *In Proceedings of the 17th annual Symposium on Computational Geometry (SoCG'01)*, pages 177–186, 2001.

[9] T. M. Chan. Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. *International Journal of Computational Geometry and Applications*, 12:67–85, 2002.

[10] T. M. Chan. Faster core-set constructions and data stream algorithms in fixed dimensions. *Computational Geometry: Theory and Applications*, 35:20–35, 2006.

[11] T. M. Chan. Applications of Chebyshev polynomials to low-dimensional computational geometry. *In Proceedings of the 33rd International Symposium on Computational Geometry (SoCG'17)*, 26:1–15, 2017.

[12] S. Arya, G. D. da Fonseca, and D. M. Mount. Near-optimal ε-kernel construction and related problems. *In Proceedings of the 33rd International Symposium on Computational Geometry (SoCG'17)*, 10:1–15, 2017.