









# Minimum-Width Double-Slabs and Widest Empty Slabs in High Dimensions

Taehoon Ahn<sup>1</sup>, Chaeyoon Chung<sup>1</sup>, Hee-Kap Ahn<sup>2</sup>, Sang Won Bae<sup>3</sup>,  
Otfried Cheong<sup>4</sup>, and Sang Duk Yoon<sup>5</sup>

<sup>1</sup> Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang, Korea

{slloth, chaeyoon17}@postech.ac.kr

<sup>2</sup> Graduate School of Artificial Intelligence, Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang, Korea

heekap@postech.ac.kr

<sup>3</sup> Division of Computer Science and Engineering, Kyonggi University, Suwon, Korea

swbae@kgu.ac.kr

<sup>4</sup> SCALGO, Aarhus, Denmark

otfried@scalgo.com

<sup>5</sup> Department of Service and Design Engineering, Sungshin Women's University, Seoul, Korea

sangduk.yoon@sungshin.ac.kr

**Abstract.** A *slab* in  $d$ -dimensional space  $\mathbb{R}^d$  is the set of points enclosed by two parallel hyperplanes. We consider the problem of finding an optimal pair of parallel slabs, called a *double-slab*, that covers a given set  $P$  of  $n$  points in  $\mathbb{R}^d$ . We address two optimization problems in  $\mathbb{R}^d$  for any fixed dimension  $d \geq 3$ : the *minimum-width double-slab* problem, in which one wants to minimize the maximum width of the two slabs of the resulting double-slab, and the *widest empty slab* problem, in which one wants to maximize the gap between the two slabs. Our results include the first nontrivial exact algorithms that solve the former problem for  $d \geq 3$  and the latter problem for  $d \geq 4$ .

**Keywords:** computational geometry · hyperplane · slab · double-slab · widest empty slab · minimum-width double-slab

## 1 Introduction

A *slab* in  $d$ -dimensional space  $\mathbb{R}^d$  is the set of points enclosed by two parallel hyperplanes. A *double-slab* is a pair of parallel slabs, that is, all four hyperplanes

C. Chung, T. Ahn, and H.-K. Ahn were supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00905, Software Star Lab (Optimal Data Structure and Algorithmic Applications in Dynamic Geometric Environment)) and (No. 2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH)). T. Ahn, S.W. Bae, and C. Chung were supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00251168).

bounding the two slabs are parallel. We consider the problem of finding an optimal double-slab that covers a given set  $P$  of  $n$  points in  $\mathbb{R}^d$ , for any fixed dimension  $d \geq 3$ . We consider two optimization problems. In the first problem, which we call the *minimum-width double-slab* problem, we wish to minimize the *width* of the resulting double-slab, where the width of a double-slab is defined as the maximum width of its two slabs. In the second problem, which we call the *widest empty slab* problem, we wish to maximize the *gap* of the double-slab, which is defined as the distance between the two slabs.

The two optimization problems concerning double-slabs extend and/or generalize several fundamental geometric problems, which have been extensively studied mostly in  $\mathbb{R}^2$  or  $\mathbb{R}^3$ , and sometimes in higher dimensions.

The *width* of a point set  $P$  in  $\mathbb{R}^d$  is defined as the minimum width of a slab that covers  $P$ . The width is considered one of the most fundamental extent measures describing a point set, together with the diameter and the radius. The width of a set of  $n$  points in the plane can be computed easily in optimal  $O(n \log n)$  time [29]. If the convex hull of  $P$  is already given, this can be improved to  $O(n)$  time using rotating calipers [31]. For  $d = 3$ , Houle and Toussaint [23] presented an  $O(n^2)$ -time algorithm, and the first subquadratic-time algorithm was given by Chazelle et al. [13]. The currently best algorithm by Agarwal and Sharir runs in  $O(n^{3/2+\epsilon})$  expected time [2]. In higher dimensions, Chan [9] discussed a simple formulation that reduces the width problem to searching a  $(d + 1)$ -dimensional convex polytope, resulting in an  $O(n^{\lceil d/2 \rceil})$ -time exact algorithm for  $d \geq 4$ . Efficient approximation schemes are also known [1, 9].

By definition, computing the width is equivalent to fitting a line for  $d = 2$  (a plane for  $d = 3$ , or in general a hyperplane) to the given point set  $P$ . In particular, for  $d = 2$ , this problem is also known as the *line center* problem. For  $k \geq 1$ , the *k-line center* problem asks to find a set of  $k$  lines such that the maximum distance of a point in  $P$  to its nearest line center is minimized. Like the *k-center* problem, this problem is known to be NP-hard when  $k$  is part of the input, even in the plane  $\mathbb{R}^2$  [28]. In spite of its fundamental position in theory and applications, little is known about exact algorithms for the *k-line center* problem. Even for  $d = 2$ , efficient algorithms are known only for  $k \leq 2$ : algorithms with running time  $O(n^2 \log^2 n)$  were found in the 1990s [20, 25] and there has been no improvement since then. For  $d \geq 3$ , we are not aware of any nontrivial exact algorithm to compute a *k-line center* for  $k \geq 2$  in the literature. Agarwal et al. [5] presented an efficient approximation algorithm for any  $k \geq 1$  and  $d \geq 2$ . Recently, some results on constrained variants of the *k-line center* problem in the plane  $\mathbb{R}^2$  have been published: Bae [6] showed that a *parallel 2-line center* can be computed in  $O(n^2)$  time, where the two lines are restricted to be parallel. Das et al. [17] presented an approximation algorithm for orthogonal line centers and Chung et al. [15] studied a variant of parallel *k-line centers* for  $k \geq 2$  considering gaps between the induced clusters.

The generalization to higher dimensions is the *k-hyperplane center* problem. Given a set  $P$  of  $n$  points in  $\mathbb{R}^d$ , we want to find  $k$  hyperplanes that minimize the maximum distance from a point in  $P$  to its nearest hyperplane. This is

equivalent to finding  $k$  slabs of minimum width that cover  $P$ . This problem has been studied in the context of the *projective clustering* problem and, except for the variants in  $\mathbb{R}^2$  mentioned above, only approximation algorithms are known (see e.g. [4] and the references therein). As discussed in [4] and [5], an exact algorithm with running time  $n^{O(dk)}$  can be easily obtained for both the  $k$ -line center and the  $k$ -hyperplane center problems. To our best knowledge, no better algorithms are known, not even for small constants  $k \geq 2$  and  $d \geq 3$ .

Our first problem, namely the minimum-width double-slab problem, can be seen as a constrained variant of the 2-hyperplane center problem, in which the two hyperplane centers must be parallel, while it extends and generalizes the fundamental geometric problems mentioned above. We give the first exact algorithm for this problem.

Covering  $P$  by a double-slab induces two clusters on  $P$ . In some applications, such as in *obnoxious facility location* [16], it is considered more important to achieve a maximum possible separation between the resulting clusters of  $P$ . This motivates us to find a double-slab cover of  $P$  with the maximum possible gap, or a *widest empty slab* through  $P$ . In the planar case  $d = 2$ , this problem is well known as the *widest empty corridor* problem. Houle and Maciel [22] presented the first  $O(n^2)$ -time algorithm for the widest empty corridor problem. Many variants of the problem have been studied since, including corridors containing at most  $k$  input points [10, 24, 30], dynamic maintenance of corridors [24, 30], and more. Díaz-Báñez et al. [18] studied the widest empty slab problem in  $\mathbb{R}^3$ , and presented an  $O(n^3)$ -time algorithm. We are not aware of any known algorithm for the widest empty slab problem in  $\mathbb{R}^d$  for any fixed dimension  $d \geq 4$ . We present the first exact algorithm.

We summarize our results as follows:

- (1) We solve the minimum-width double-slab problem in  $O(n^d)$  time for  $d = 3, 4$ , and in  $O(n^{d+1})$  time for any fixed dimension  $d \geq 5$ .
- (2) We solve the widest empty slab problem in  $O(n^d \log n)$  time for any fixed dimension  $d \geq 4$ .

We characterize combinatorial properties of optimal solutions to both problems, and find the optimal solution by efficiently enumerating all candidates. We make a heavy use of geometric duality, mapping points in  $\mathbb{R}^d$  into non-vertical hyperplanes in  $\mathbb{R}^d$ , and known algorithms and data structures for hyperplanes and their arrangement in high dimensions.

Due to page limit, most proofs are omitted and will be seen in a full version.

## 2 Preliminaries

Let  $d \geq 2$  be an arbitrary fixed dimension. We consider the  $d$ -dimensional Euclidean space  $\mathbb{R}^d$  with the  $d$  coordinate axes, called the  $x_1, x_2, \dots, x_d$ -axes. We consider the  $x_d$ -axis as the *vertical* direction. We also treat  $\mathbb{R}^d$  as a vector space equipped with the standard inner product and the induced Euclidean norm  $\|\cdot\|$ . Hence, for any two points  $p, q \in \mathbb{R}^d$ , the length of segment  $pq$  is  $\|p - q\|$ . For

any subset  $A \subset \mathbb{R}^d$ , let  $\text{aff}(A)$  be the affine hull of  $A$ , that is, the intersection of all affine subspaces containing  $A$ .

For any affine subspace  $A$  of  $\mathbb{R}^d$ , we denote by  $A^\perp$  the *orthogonal complement* of  $A$  in  $\mathbb{R}^d$ , the vector subspace consisting of all vectors in  $\mathbb{R}^d$  that are orthogonal to every vector  $a - a_0$  for  $a, a_0 \in A$ . Note that  $\dim(A) + \dim(A^\perp) = d$ . Let  $\pi_A: \mathbb{R}^d \rightarrow A$  denote the orthogonal projection onto  $A$ . In particular, if  $A = \mathbb{R}^{d-1} = \{x_d = 0\}$ , then we simply write  $\pi = \pi_{\mathbb{R}^{d-1}}$ . The projection  $\pi$  drops the  $x_d$ -coordinate, so that for any  $(a_1, \dots, a_d) \in \mathbb{R}^d$ , we have  $\pi(a_1, \dots, a_{d-1}, a_d) = (a_1, \dots, a_{d-1}) \in \mathbb{R}^{d-1}$ .

*Hyperplanes and orientations.* We call a  $k$ -flat *vertical* if it is parallel to the  $x_d$ -axis. A *hyperplane* in  $\mathbb{R}^d$  is a  $(d - 1)$ -flat. Any non-vertical hyperplane  $h \subset \mathbb{R}^d$  can be seen as the graph of a  $(d - 1)$ -variate linear function  $h: \mathbb{R}^{d-1} \rightarrow \mathbb{R}$ :  $h: x_d = u_1x_1 + u_2x_2 + \dots + u_{d-1}x_{d-1} + b$ , for some  $u = (u_1, u_2, \dots, u_{d-1}) \in \mathbb{R}^{d-1}$  and  $b \in \mathbb{R}$ . We call  $u$  the *orientation* of  $h$ , and  $b$  its *displacement*. Any non-vertical hyperplane  $h$  can be uniquely determined by an orientation  $u$  and a displacement  $b$  in this way, and we have the following.

**Observation 1.** *Let  $h$  be a non-vertical hyperplane in  $\mathbb{R}^d$  with orientation  $u \in \mathbb{R}^{d-1}$ . Then, the vector  $(u, -1) \in \mathbb{R}^d$  is a normal of  $h$ .*

In particular, hyperplanes with equal orientation are parallel. We can thus identify the space of possible orientations of non-vertical hyperplanes with  $\mathbb{R}^{d-1}$ . For any orientation  $u \in \mathbb{R}^{d-1}$ , let  $\theta(u)$  be the angle between the vector  $(u, -1) \in \mathbb{R}^d$  and the negative direction of the  $x_d$ -axis. For any two parallel hyperplanes  $h_1$  and  $h_2$ , let  $w(h_1, h_2)$  denote the distance between  $h_1$  and  $h_2$ . We observe that  $w(h_1, h_2) = |b_1 - b_2| \cdot \cos(\theta(u)) = \frac{|b_1 - b_2|}{\|(u, -1)\|}$ , where  $u$  denotes their common orientation and  $b_i$  the displacement of  $h_i$ , for  $i = 1, 2$ .

A *slab*  $S$  is the closure of the region between two parallel hyperplanes  $h_1$  and  $h_2$ , denoted by  $S = (h_1, h_2)$ . The *width* of  $S$ , denoted by  $w(S)$ , is  $w(h_1, h_2)$ . A *double-slab*  $D = (S_1, S_2)$  is the union of two disjoint parallel slabs  $S_1$  and  $S_2$ . The *width* of  $D$ , denoted by  $w(D)$ , is  $\max\{w(S_1), w(S_2)\}$ . We say that a slab or double-slab is *vertical* if its defining hyperplanes are vertical. For any non-vertical slab or double-slab, its *orientation* is the orientation of its defining hyperplanes.

*Duality.* We recall the classic point-to-hyperplane duality transform: for any point  $p = (a_1, a_2, \dots, a_d) \in \mathbb{R}^d$ , the dual transformation maps  $p$  into its dual hyperplane  $p^*: x_d = a_1x_1 + \dots + a_{d-1}x_{d-1} - a_d$ . Conversely, any non-vertical hyperplane  $h \subset \mathbb{R}^d$  is mapped to a point  $h^* \in \mathbb{R}^d$ . It is well known that the duality transform  $a \mapsto a^*$  preserves the point-hyperplane incidence relation and the vertical order among points and hyperplanes [8].

**Observation 2.** *Let  $S = (h_1, h_2)$  be a non-vertical slab with orientation  $u$ .*

- The segment  $h_1^*h_2^*$  is vertical, that is, it is parallel to the  $x_d$ -axis in  $\mathbb{R}^d$ .
- It holds that  $w(S) = \|h_1^* - h_2^*\| \cos(\theta(u))$ .
- The first  $d - 1$  coordinates of  $h_1^*$  and  $h_2^*$  are equal to those of  $u$ , that is,  $\pi(h_1^*) = \pi(h_2^*) = u$ .

– For any point  $p \in \mathbb{R}^d$ ,  $p \in S$  if and only if the segment  $h_1^*h_2^*$  intersects  $p^*$ .

Therefore, there is a one-to-one correspondence between non-vertical slabs and vertical segments under the duality transform.

*Arrangement of Hyperplanes.* Let  $H$  be a set of  $n$  hyperplanes in  $\mathbb{R}^d$ . Consider the arrangement  $\mathcal{A}(H)$  of these  $n$  hyperplanes. We introduce some algorithms and data structures for hyperplane arrangements that we will be using.

The arrangement  $\mathcal{A}(H)$  consists of  $O(n^d)$  faces for  $d \geq 2$  [21]. The upper and lower envelopes of  $H$ , denoted by  $\mathcal{U}(H)$  and  $\mathcal{L}(H)$ , correspond to the convex hull of points that are dual to  $H$ , so their complexity is bounded by  $O(n^{\lfloor d/2 \rfloor})$ . It is well known that  $\mathcal{A}(H)$  can be computed in  $O(n^d)$  time [21] and the envelopes  $\mathcal{U}(H)$  and  $\mathcal{L}(H)$  can be computed in  $O(n \log n + n^{\lfloor d/2 \rfloor})$  time, using any optimal convex hull algorithm, such as the one by Chazelle [12].

The zone  $Z(h; H)$  of another hyperplane  $h$  in the arrangement of  $H$  is the set of all cells in  $\mathcal{A}$  intersected by  $h$  and their incident faces. Edelsbrunner et al. [19] showed that the complexity of the zone  $Z(h; H)$  is  $\Theta(n^{d-1})$ .

**Lemma 1. (de Berg et al. [7]).** *For any fixed  $d \geq 2$ , the zone of a hyperplane  $h$  in an arrangement  $\mathcal{A}(H)$  of  $n$  hyperplanes  $H$  can be computed in  $O(n^{d-1} + n \log n)$  time, without computing the whole arrangement  $\mathcal{A}$ .*

In our algorithms, it is often required to test the feasibility of a candidate slab. The following query structure for point location in the arrangement will be used for our purpose.

**Lemma 2. (Chazelle [11] and Matoušek [26]).** *A set  $H$  of  $n$  hyperplanes in  $\mathbb{R}^d$  can be preprocessed in  $O(n^d / \log^d n)$  time into a data structure of size  $O(n^d / \log^d n)$  that answers the following query in  $O(\log n)$  time: Given a query point  $q \in \mathbb{R}^d$ , locate the face in  $\mathcal{A}(H)$  that contains  $q$  and count the number of hyperplanes in  $H$  above  $q$ .*

By the duality transform, this data structure can answer the *half-space counting query* [3]. Thus, given a set  $P$  of  $n$  points in  $\mathbb{R}^d$ , by using the query structure in Lemma 2 for the set of  $n$  hyperplanes dual to points in  $P$ , we can count the number of points in  $P$  contained in a query slab  $S = (h_1, h_2)$  in  $O(\log n)$  time, according to Observation 2.

*General Position Assumption.* In the following,  $P$  will be a set of  $n$  input points in  $\mathbb{R}^d$ . For simplicity, we will assume that  $P$  is in general position, meaning that no hyperplane in  $\mathbb{R}^d$  contains more than  $d$  points of  $P$  or, equivalently, that any  $d + 1$  points in  $P$  are affinely independent. Hence, for any subset  $Q \subseteq P$  of  $k \leq d + 1$  points, its affine hull  $\text{aff}(Q)$  is a  $(k - 1)$ -flat in  $\mathbb{R}^d$ . This also implies that the orthogonal complement  $(\text{aff}(Q))^\perp$  is of dimension  $d - k + 1$ . Throughout the paper, we often discuss the intersection of two orthogonal complements  $V = (\text{aff}(Q_1))^\perp \cap (\text{aff}(Q_2))^\perp$  for two nonempty disjoint subsets  $Q_1, Q_2 \subset P$  with  $k = |Q_1| + |Q_2| \leq d + 1$ . We shall call  $V$  the (linear) *subspace orthogonal to both*  $\text{aff}(Q_1)$  and  $\text{aff}(Q_2)$ . The general position also implies that the subspace  $V$  is always of dimension  $d - k + 2$  for any two subsets  $Q_1, Q_2 \in P$ .

### 3 Widest Empty Slabs

Let us call a slab  $S$  *empty* if it contains no point of  $P$  in its interior while separating  $P$  into two nonempty subsets. Our goal in this section is to compute an empty slab  $S$  of maximum width. The following is an easy observation on empty slabs.

**Lemma 3.** *Suppose that  $S$  is a maximum-width empty slab for  $P$  in  $\mathbb{R}^d$  for  $d \geq 3$ . Let  $h \subset \mathbb{R}^d$  be any hyperplane parallel to the normal of  $S$ . Then,  $\pi_h(S)$  is a maximum-width empty slab for  $\pi_h(P)$  in the  $(d - 1)$ -dimensional space  $h$ .*

Lemma 3 allows us to find the widest *vertical* empty slab by projecting the point set to  $\mathbb{R}^{d-1}$  and solving the problem there. We can therefore concentrate on finding the widest *non-vertical* empty slab, and in the remainder of this section all slabs will be non-vertical. We represent a non-vertical slab  $S$  by a pair  $(h_1, h_2)$  of hyperplanes such that  $h_1$  is above  $h_2$ .

We prove the following characterization of optimal empty slabs.

**Lemma 4.** *For any  $d \geq 2$ , if  $S = (h_1, h_2)$  is an empty slab of maximum width, then one of the following must hold:*

- (i) *At least  $d + 1$  points lie on the boundary of  $S$ , that is,  $|P \cap (h_1 \cup h_2)| \geq d + 1$ .*
- (ii) *The normal vector to  $S$  is parallel to  $\text{aff}(P \cap (h_1 \cup h_2))$ .*

*Proof.* Let  $S = (h_1, h_2)$  be a maximum-width empty slab for  $P$ , and let  $k := |P \cap (h_1 \cup h_2)|$  be the number of points in  $P$  that lie on the boundary of  $S$ . Clearly at least one point of  $P$  lies on each of  $h_1$  and  $h_2$ , so  $k \geq 2$ . Let  $u \in \mathbb{R}^{d-1}$  be the orientation of  $S$ . Then,  $\hat{u} := (u, -1)$  is a normal vector of  $S$  by Observation 1.

We will prove by induction on  $k$  and  $d$  that whenever  $k \leq d$ , the statement in the second case holds, that is, the normal  $\hat{u}$  is parallel to  $\text{aff}(P \cap (h_1 \cup h_2))$ .

Consider first the case  $k = 2$ . The planar case of the statement was proven by Houle and Maciel [22], see also Janardan and Preparata [24, Theorem 2.1]. Consider now  $d > 2$  and assume the claim holds in  $\mathbb{R}^{d-1}$  (for  $k = 2$ ). Let  $q_1, q_2 \in P$  be the two points such that  $q_1 \in h_1$  and  $q_2 \in h_2$ , and suppose for a contradiction that  $\hat{u}$  is not parallel to  $\ell := \text{aff}(P \cap (h_1 \cup h_2))$ . In this case,  $\ell$  is the line through  $q_1$  and  $q_2$ . Let  $\ell'$  be the line parallel to  $\hat{u}$  going through  $q_1$ . The two lines  $\ell$  and  $\ell'$  make a positive angle  $\phi > 0$ . We consider a hyperplane  $h$  containing both lines  $\ell$  and  $\ell'$ . Since the normal of  $h$  is parallel to  $S$ , Lemma 3 implies that  $\pi_h(S)$  is a maximum-width empty slab for  $\pi_h(P)$  in  $h$ . The two points  $q_1$  and  $q_2$  lie in  $h$  and are still the only two points of  $P$  on the boundary of  $\pi_h(S)$ . The normal of  $\pi_h(S)$  is identical to  $\hat{u}$  and still makes a positive angle  $\phi$  with  $\ell$ , a contradiction to the optimality of  $\pi_h(S)$ . See Fig. 1(a). It follows that the claim holds for  $k = 2$  in any dimension.

Consider now the general case  $d > 2$  and  $2 < k \leq d$ , and assume that the claim holds for all smaller dimensions and smaller values of  $k$ . In this case there are at least two points in  $P \cap h_1$  or  $P \cap h_2$ . Without loss of generality, we let  $|P \cap h_1| \geq 2$ . Let  $f$  be the  $(k - 1)$ -flat  $f = \text{aff}(P \cap (h_1 \cup h_2))$ . We pick two points  $q_1, q_2 \in P \cap h_1$ . Let  $\ell$  be the line through  $q_1$  and  $q_2$ , and let  $h$  be the hyperplane

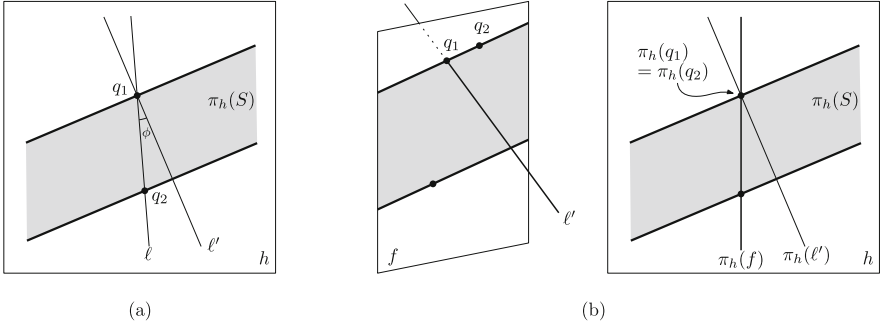


Fig. 1. Illustration to the proof of Lemma 4.

normal to  $\ell$  containing  $q_1$ . Since the normal of  $h$  is parallel to  $S$ , we again apply Lemma 3 to conclude that  $\pi_h(S)$  is a maximum-width empty slab for  $\pi_h(P)$  in  $h$ . Since  $\pi_h(q_1) = \pi_h(q_2)$ , there are  $k - 1$  points of  $\pi_h(P)$  lying on the boundary of  $\pi_h(S)$ . By the inductive assumption, the normal  $\hat{u}$  of  $\pi_h(S)$  is parallel to the  $(k - 2)$ -flat  $f' = \text{aff}(\pi_h(P) \cap (\pi_h(h_1) \cup \pi_h(h_2)))$ . Since our projection direction lies in  $f$ , we have  $f' \subset f$ , and so  $\hat{u}$  is parallel to  $f$ , completing the inductive step. See Fig. 1(b) for an illustration when  $d = 3$  and  $k = 3$ .  $\square$

We will now describe our algorithm that computes a maximum-width empty slab for a given set  $P$  of  $n$  points in  $\mathbb{R}^d$ . It enumerates all candidate slabs that satisfy one of the two conditions described in Lemma 4. We will show that the number of candidates is bounded by  $O(n^d)$  and we will spend  $O(\log n)$  time per candidate for the emptiness test.

Let  $P^*$  be the set of  $n$  hyperplanes dual to the points  $P$ , and let  $\mathcal{A} = \mathcal{A}(P^*)$  be their arrangement. We first build the query structure of Lemma 2 for  $\mathcal{A}$ . We will separately enumerate all candidates that fall in case (i) and those that fall in case (ii) of Lemma 4.

### 3.1 Case (i)

Let us call an empty slab  $S$  a *candidate slab* if it satisfies the condition of case (i) in Lemma 4. In dual space, Observation 2 immediately implies the following.

**Lemma 5.** *A candidate slab corresponds to a maximal vertical segment contained in a  $d$ -dimensional cell of  $\mathcal{A}$ , whose endpoints lie in the relative interior<sup>1</sup> of two faces  $f_1$  and  $f_2$  with  $\dim(f_1) + \dim(f_2) \leq d - 1$ .*

We call such a vertical segment a *candidate stick*. Our algorithm collects all candidate sticks from the arrangement  $\mathcal{A}$ , computes the width of the corresponding slab for each candidate stick, and returns one that maximizes this width.

The candidate sticks are closely related to a vertical decomposition of the arrangement of hyperplanes. Chazelle and Friedman [14] showed that the number

<sup>1</sup> Note that the relative interior of a 0-flat (a point) is by definition the point itself.

of candidate sticks is  $O(n^d)$  and argued that they can be computed by overlaying the orthogonal projections of all the faces incident to each  $d$ -dimensional cell of  $\mathcal{A}$  onto the horizontal hyperplane  $\{x_d = 0\}$ . They also described an algorithm that computes all candidate sticks in  $O(n^{\lfloor 3d/2 \rfloor} \log^{\lfloor d/2 \rfloor} n)$  time,<sup>2</sup> which is too slow for our purpose.

Consider a candidate stick  $s$  between two faces  $f_1$  and  $f_2$  of  $\mathcal{A}$ . By Lemma 5, their dimensions sum up to at most  $d - 1$ , so the smaller dimension of the two, say  $\dim(f_2)$ , is at most  $\lfloor (d-1)/2 \rfloor$ . Let  $k := \dim(f_2) + 1$ . Then, we have  $\dim(f_1) \leq d - k$ , and so  $f_1$  lies on (is a subface of) a face  $f'$  of  $\mathcal{A}$  of dimension  $\dim(f') = d - k$ . The face  $f'$  is a face on the  $(d - k)$ -flat  $f_Q := \bigcap_{q \in Q} q^*$  for some subset  $Q \in P$  with  $|Q| = k$ . Let now  $\hat{f}_Q$  be the  $(d - k + 1)$ -flat spanned by  $f_Q$  and the vertical direction, and observe that  $s$  lies in  $\hat{f}_Q$ , with one endpoint in  $f_Q$ .

Every hyperplane  $q^*$  intersects  $\hat{f}_Q$  in a  $(d - k)$ -flat  $q'$ . The face  $f_2$  of  $\mathcal{A}$  has  $\dim(f_2) = k - 1$  and therefore lies on the intersection of  $d - k + 1$  hyperplanes of  $\mathcal{A}$ . The intersection of these hyperplanes with the  $(d - k + 1)$ -flat  $\hat{f}_Q$  is therefore zero-dimensional, that is, a vertex of the arrangement  $\mathcal{A}_Q$  of the  $(d - k)$ -flats  $q'$ , for  $q \in Q$ , inside the  $(d - k + 1)$ -flat  $\hat{f}_Q$ .

To summarize, for our candidate stick  $s$  there is a subset  $Q \subset P$  of size  $|Q| = k$ , for some  $1 \leq k \leq \lfloor (d - 1)/2 \rfloor + 1 = \lceil d/2 \rceil$ , such that one endpoint of  $s$  is a vertex  $v$  of the arrangement  $\mathcal{A}_Q$  in  $\hat{f}_Q$  and the other endpoint is the vertical projection of  $v$  on the  $(d - k)$ -flat  $f_Q$ . No hyperplane  $q'$  crosses the relative interior of  $s$ , so the entire segment lies in the zone of  $f_Q$  in the arrangement  $\mathcal{A}_Q$ .

Our algorithm enumerates all possibilities: We consider all non-empty subsets  $Q \subset P$  of size at most  $\lceil d/2 \rceil$ . For each  $Q$ , we compute the flat  $f_Q$  and its vertical extension  $\hat{f}_Q$ , and intersect all hyperplanes with  $\hat{f}_Q$ . We then compute the zone  $Z$  of  $f_Q$  in the arrangement  $\mathcal{A}_Q$ , and for each vertex  $v$  of  $Z$ , generate a vertical segment connecting  $v$  with its vertical projection on  $f_Q$ . Finally, we test whether the segment actually is a candidate stick by verifying whether its relative interior intersects any hyperplane of  $\mathcal{A}$  using the data structure of Lemma 2.

There are  $\binom{n}{k} = O(n^k)$  different subsets of  $k$  points. Computing the zone of  $n$  hyperplanes in the  $(d - k + 1)$ -dimensional space  $\hat{f}_Q$  takes time  $O(n^{d-k} + n \log n)$  time by Lemma 1. By the zone theorem [19], the zone has at most  $O(n^{d-k})$  vertices, so we can test all the resulting vertical segments in time  $O(n^{d-k} \log n)$ . Summing over all  $O(n^k)$  subsets  $Q$  of size  $k$  results in time  $O(n^d \log n + n^{k+1} \log n)$ , and summing over  $k$  from 1 to  $\lceil d/2 \rceil$  results in total time  $O(n^d \log n)$ .

### 3.2 Case (ii)

We now turn to the second case. We will enumerate *all* slabs  $S$  (not just empty ones) that are determined by a subset  $Q$  of at most  $d$  points of  $P$  in the following way: The intersection of the boundary of  $S$  with  $P$  is exactly  $Q$ , and the normal

---

<sup>2</sup> In fact they construct a vertical decomposition of the entire arrangement.



of  $S$  is parallel to  $\text{aff}(Q)$ . We then test whether the slab is actually empty using the data structure of Lemma 2, and return an empty slab maximizing the width.

The following lemma shows that given the boundary points, the unique slab satisfying the condition can be found in constant time.

**Lemma 6.** *Let  $Q_1, Q_2 \subset P$  be two disjoint subsets such that  $2 \leq |Q_1| + |Q_2| \leq d$ . There exists a unique orientation  $u$  of a slab  $S = (h_1, h_2)$  such that  $P \cap h_1 = Q_1$ ,  $P \cap h_2 = Q_2$ , and the normal of  $S$  is parallel to  $\text{aff}(Q_1 \cup Q_2)$ . Moreover, such a unique orientation can be computed in  $O(d^3) = O(1)$  time.*

Let  $Q_1, Q_2 \subset P$  be a pair of disjoint subsets of  $P$  with  $2 \leq |Q_1 \cup Q_2| \leq d$ . In constant time we can compute the unique orientation described in Lemma 6, denoted by  $u(Q_1, Q_2)$ , and construct a slab  $S = (h_1, h_2)$  with orientation  $u(Q_1, Q_2)$  such that  $P \cap h_1 = Q_1$  and  $P \cap h_2 = Q_2$ . We can then test in  $O(\log n)$  time if the relative interior of  $S$  is empty using the data structure of Lemma 2. We return the widest slab among all slabs passing the test.

We run the above procedure for all valid pairs  $(Q_1, Q_2)$ . The number of such pairs is bounded by  $\sum_{2 \leq k \leq d} \binom{n}{k} \cdot 2^k = O(n^d)$ , since  $d$  is a constant. Hence, the total running time for case (ii) is also bounded by  $O(n^d \log n)$ .

**Theorem 1.** *For any constant dimension  $d \geq 4$ , a widest empty slab for a set of  $n$  points in  $\mathbb{R}^d$  can be computed in  $O(n^d \log n)$  time.*

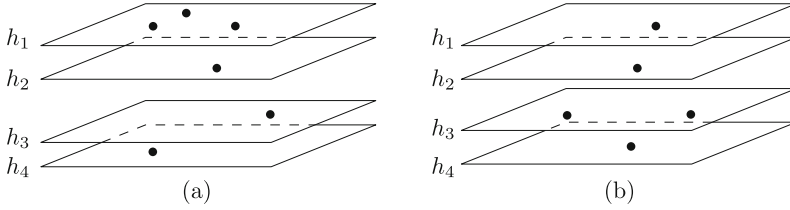
Our algorithm can be implemented faster for  $d \leq 3$ , because the emptiness of a candidate slab can be tested in  $O(1)$  amortized time. This results in algorithms with  $O(n^2)$  and  $O(n^3)$  time for  $d = 2$  and  $d = 3$ , respectively, matching the previously known time bounds [18, 22]. The space requirement of our algorithm, however, is larger.

## 4 Minimum-Width Double-Slabs

In this section, we present algorithms that compute a minimum-width double-slab enclosing  $P$  in  $\mathbb{R}^d$  for  $d \geq 3$ . Throughout this section, we assume that there exists a minimum-width double-slab enclosing  $P$  that is *non-vertical*. As in the previous section, if this is not the case, we can solve the problem by reducing it to an instance in  $\mathbb{R}^{d-1}$  after orthogonally projecting  $P$  onto  $\mathbb{R}^{d-1}$ . In addition, we also assume that  $P$  consists of at least  $d+2$  points, since the problem becomes trivial when  $|P| \leq d+1$ .

Let  $P^*$  be the set of  $n$  hyperplanes dual to points in  $P$ ,  $\mathcal{A} = \mathcal{A}(P^*)$  be their arrangement,  $\mathcal{L} = \mathcal{L}(P^*)$  be the lower envelope of  $\mathcal{A}$ , and  $\mathcal{U} = \mathcal{U}(P^*)$  be the upper envelope of  $\mathcal{A}$ . We represent any non-vertical double-slab  $D$  by a 4-tuple  $(h_1, h_2, h_3, h_4)$  of hyperplanes such that  $h_i$  is above  $h_{i+1}$  in  $\mathbb{R}^d$  for each  $i = 1, 2, 3$ .

It is obvious that there exists a minimum-width double-slab enclosing  $P$  such that at least one point in  $P$  lies on each of its four hyperplanes. The following lemma describes a characterization of optimal double-slabs for our problem. See also Fig. 2.



**Fig. 2.** Two possible configurations of minimum-width double-slabs  $D = (h_1, h_2, h_3, h_4)$  in  $\mathbb{R}^3$ . (a) Case (i) of Lemma 7: There are six points on the four planes. (b) Case (iii): There are five points on the four planes, while we have  $w(h_1, h_2) = w(h_3, h_4)$ .

**Lemma 7.** For any  $d \geq 2$ , if  $D = (h_1, h_2, h_3, h_4)$  is a minimum-width double-slab enclosing  $P$  such that  $|P \cap h_i| \geq 1$  for each  $i \in \{1, 2, 3, 4\}$ . Then, the following conditions must hold:

- $\sum_i |P \cap h_i| \geq d + 2$ .
- Exactly one of the following cases holds:
  - (i)  $w(h_1, h_2) > w(h_3, h_4)$  and  $|P \cap h_1| + |P \cap h_2| \geq d + 1$ .
  - (ii)  $w(h_1, h_2) < w(h_3, h_4)$  and  $|P \cap h_3| + |P \cap h_4| \geq d + 1$ .
  - (iii)  $w(h_1, h_2) = w(h_3, h_4)$ .

Consider any minimum-width double-slab  $D = (h_1, h_2, h_3, h_4)$  enclosing  $P$  that satisfies the conditions described in Lemma 7. Let  $Q_i := P \cap h_i$  for  $i = 1, 2, 3, 4$ . Since the outer slab  $S_{\text{out}} = (h_1, h_4)$  also encloses  $P$ , it is obvious that a portion of  $\bigcap_{q \in Q_1} q^*$  appears as a face of  $\mathcal{L}$  and a portion of  $\bigcap_{q \in Q_4} q^*$  appears in  $\mathcal{U}$  in the same way. Hence, we have  $h_1^* \in \mathcal{L}$  and  $h_4^* \in \mathcal{U}$ . On the other hand, the inner slab  $S_{\text{in}} = (h_2, h_3)$  should be empty, so the vertical segment  $h_2^* h_3^*$  crosses no hyperplane in  $P^*$ . Observe also that the dual point  $h_i^*$  for each  $i$  lies in the relative interior of a  $(d - |Q_i|)$ -face of  $\mathcal{A}$  that is a portion of the flat  $\bigcap_{q \in Q_i} q^*$ .

We will often discuss the overlay  $\mathcal{M}$  of the orthogonal projections  $\pi(\mathcal{U})$  and  $\pi(\mathcal{L})$  of the envelopes  $\mathcal{U}$  and  $\mathcal{L}$  onto the hyperplane  $\{x_d = 0\} = \mathbb{R}^{d-1}$ . Each face  $f$  of the overlay  $\mathcal{M}$  is associated with a pair  $(Q_1, Q_4)$  of subsets of  $P$  such that: for any orientation  $u \in f$ , if  $(h_1, h_4)$  is the minimum-width slab of orientation  $u$  that encloses  $P$ , then it always holds that  $P \cap h_1 = Q_1$  and  $P \cap h_4 = Q_4$ .

**Lemma 8.** For any fixed  $d \geq 2$ , the overlay  $\mathcal{M}$  consists of  $O(n^{\lceil d/2 \rceil})$  many faces and can be computed in  $O(n^{\lceil d/2 \rceil} + n \log n)$  time.

In the following, we describe our algorithms for  $d \geq 3$ .

### 4.1 Three-Dimensional Case

First, we consider the case of  $d = 3$ . For any point  $q \in P$  and any orientation  $u \in \mathbb{R}^2$ , we consider the double-slab  $D_q(u) = (h_1, h_2, h_3, h_4)$  with orientation  $u$  such that  $P \subset D_q(u)$ ,  $h_1^* \in \mathcal{L}$ ,  $q' \in h_2$  (that is,  $h_2^* \in q'^*$ ) for some  $q' \in P \setminus \{q\}$ ,  $q \in h_3$  (that is,  $h_3^* \in q^*$ ), and  $h_4^* \in \mathcal{U}$ . Since  $D_q(u)$  encloses  $P$ , the point  $q'$  on  $h_2$  is

determined as the first point in  $P$  hit by translating  $h_3$  upwards or, equivalently, as the first hyperplane  $q^*$  in  $\mathcal{A}$  hit by the ray emanating downwards from  $h_3^*$ . By Lemma 7, there exist some  $q \in P$  and  $u \in \mathbb{R}^2$  such that  $D_q(u)$  is indeed a minimum-width double-slab enclosing  $P$ .

Let  $w_q(u) := w(D_q(u))$  be its width. Then, we have  $w_q(u) = \max\{\|h_1^* - h_2^*\|, \|h_3^* - h_4^*\|\} \cdot \cos(\theta(u))$ . Let  $v_q(u) := w_q(u) / \cos(\theta(u))$  be the linear part of the right side of the above equation. Observe that the function  $v_q: \mathbb{R}^2 \rightarrow \mathbb{R}$  is piecewise linear, since its values are determined by the vertical distances between faces in  $\mathcal{A}$ , while  $\cos(\theta(u)) = 1/\sqrt{\|u\|^2 + 1}$  is a continuous function over  $u \in \mathbb{R}^2$ .

**Lemma 9.** *The number of linear pieces in function  $v_q$  is  $O(n^2)$  and an explicit description of function  $v_q$  can be computed in  $O(n^2)$  time.*

By Lemma 9, we also obtain the planar subdivision  $M_q$  induced by the break-points of  $v_q$  such that  $v_q$  restricted to any face of  $M_q$  (of any dimension) is a linear function. Lemma 7 for  $d = 3$  implies the existence of a minimum-width double-slab enclosing  $P$  whose orientation lies on a vertex or an edge of  $M_q$ . Since the function  $v_q$  restricted to each face of  $M_q$  (of any dimension) is linear and thus of constant complexity, we can find an optimal double-slab constrained about  $q \in P$  in total  $O(n^2)$  time. By iterating all  $q \in P$ , we obtain the following.

**Theorem 2.** *Given a set  $P$  of  $n$  points in  $\mathbb{R}^3$ , a minimum-width double-slab enclosing  $P$  can be computed in  $O(n^3)$  time.*

### 4.2 Four-Dimensional Case

We then consider the four-dimensional case,  $d = 4$ . Consider an optimal solution  $D = (h_1, h_2, h_3, h_4)$  that satisfies the conditions of Lemma 7. We distinguish two cases: when (a)  $|P \cap h_2| = |P \cap h_3| = 1$  or (b) either  $|P \cap h_2| \geq 2$  or  $|P \cap h_3| \geq 2$ . Our goal is to find an optimal double-slab that falls into each case.

In the former case (a), we first compute the overlay  $\mathcal{M}$  of  $\pi(\mathcal{U})$  and  $\pi(\mathcal{L})$  by applying Lemma 8. Then, for each face of  $\mathcal{M}$ , consider its associated pair  $(Q_1, Q_4)$  of subsets and a minimum-width double-slab  $D$  enclosing  $P$  such that  $P \cap h_1 = Q_1$ ,  $P \cap h_4 = Q_4$ , and it falls into case (a). Since we must have  $|Q_1| + |Q_4| \geq 4$  by Lemma 7, we ignore those faces of  $\mathcal{M}$  whose associated pair  $(Q_1, Q_4)$  consists of at most three points. Since our target double-slab contains  $Q_1$  and  $Q_4$  on its outer boundary, its normal should be orthogonal both to  $\text{aff}(Q_1)$  and  $\text{aff}(Q_4)$ . Hence, we consider the subspace  $f = (\text{aff}(Q_1))^\perp \cap (\text{aff}(Q_4))^\perp$  orthogonal to both  $\text{aff}(Q_1)$  and  $\text{aff}(Q_4)$ . As discussed in Sect. 2, such a flat  $f$  exists and its dimension is always  $6 - |Q_1| - |Q_4| \leq 2$ . Hence,  $f$  is a line or a plane in  $\mathbb{R}^4$ . If  $f$  is a line, then we can find our solution in  $O(n)$  time by scanning all points in  $P$ ; if  $f$  is a plane, then we project  $P$  orthogonally onto  $f$  and solve the 2-dimensional problem instance in  $O(n^2)$  time [6]. Since  $\mathcal{M}$  consists of  $O(n^2)$  faces (Lemma 8), we can handle case (a) in a total of  $O(n^4)$  time.

Next, we consider the latter case (b). Assume without loss of generality that we have  $|Q_3| \geq 2$  in an optimal double-slab of this case. We fix a pair of points

$q, q' \in P$  and consider the case where  $q, q' \in Q_3$ . For the purpose, we consider any hyperplane  $h$  in  $\mathbb{R}^4$  orthogonal to the line through  $q$  and  $q'$ , and the orthogonal projections of  $P$  onto  $h$ . Note that  $q$  and  $q'$  are projected to a common point  $\bar{q}$  on  $h$ . We then have a problem instance in  $\mathbb{R}^3$  with the constraint that  $\bar{q}$  should lie on the third plane in the resulting slab. Therefore, we apply Lemma 9 to solve this instance in  $O(n^2)$  time. We can thus handle case (b) by iterating all pairs  $(q, q')$  of two points in  $P$ , taking  $O(n^4)$  time.

**Theorem 3.** *Given a set  $P$  of  $n$  points in  $\mathbb{R}^4$ , a minimum-width double-slab enclosing  $P$  can be computed in  $O(n^4)$  time.*

### 4.3 Algorithm for Higher Dimension

Finally, we consider higher dimensions for  $d \geq 5$ . The algorithm starts with some preprocessing, including the initialization of the point location structure of Lemma 2 and the query structure by Matoušek and Schwarzkopf [27] for ray shooting for  $\mathcal{U}$  and  $\mathcal{L}$ . The cost of preprocessing is not more than  $O(n^d)$  time. In the main part of the algorithm, we handle each of the three cases described in Lemma 7.

*Cases (i) and (ii).* First, we handle double-slabs  $D = (h_1, h_2, h_3, h_4)$  enclosing  $P$  of case (i):  $w(h_1, h_2) > w(h_3, h_4)$  and  $|P \cap h_1| + |P \cap h_2| \geq d + 1$ . For the purpose, we consider those  $D = (h_1, h_2, h_3, h_4)$  with the additional constraint that only one fixed point  $q \in P$  lies on  $h_2$ .

**Lemma 10.** *Let  $d \geq 2$  be any fixed integer and  $q \in P$  be any fixed point. A minimum-width double-slab  $D = (h_1, h_2, h_3, h_4)$  enclosing  $P$  such that  $w(h_1, h_2) > w(h_3, h_4)$  and  $P \cap h_2 = \{q\}$  can be computed in  $O(n^{\lfloor d/2 \rfloor} \log n)$  time, provided the query structure of Lemma 2.*

For each  $1 \leq k \leq d$ , consider an optimal double-slab  $D$  of this case such that  $|P \cap h_2| = k$ . Such an optimal solution  $D$  can be computed as follows: For each subset  $Q_2 \subset P$  with  $|Q_2| = k$ , let  $f := (\text{aff}(Q_2))^\perp$  be the orthogonal complement of  $\text{aff}(Q_2)$ . Note that  $f$  is a  $(d - k + 1)$ -flat, as discussed in Sect. 2. We then orthogonally project  $P$  onto  $f$  by the projection operator  $\pi_f$ , and find an optimal double-slab  $D' = (h'_1, h'_2, h'_3, h'_4)$  of case (i) with the additional constraint that  $\pi_f(P) \cap h'_2 = \pi_f(Q_2)$ . This constrained problem can be solved by Lemma 10 if  $1 \leq k \leq d - 1$ , since the projection  $\pi_f$  maps all points in  $Q_2$  to a common point in  $f$  and thus  $\pi_f(Q_2)$  consists of a single point. Since there are  $\binom{n}{k}$  different choices of  $Q_2$  and we spend  $O(n^{\lfloor (d-k+1)/2 \rfloor} \log n)$  time per each (Lemma 10), it takes  $O(n^d \log n)$  time in total over all  $1 \leq k \leq d - 1$ .

In case of  $k = d$ , the dimension of flat  $f$  is 1, so it gives us a unique orientation  $u \in \mathbb{R}^{d-1}$ . Hence, this case can be solved in  $O(\log n)$  time as follows: Using the ray shooting queries for  $\mathcal{L}$  and  $\mathcal{U}$  [27], we locate the points  $h_1^*$  and  $h_4^*$  on  $\mathcal{L}$  and  $\mathcal{U}$ , respectively, at  $u$  in  $O(\log n)$  time, and then test if the vertical segment with top endpoint at  $h_4^*$  and length  $\|h_1^* - h_4^*\|$  crosses the correct number of hyperplanes in

A. This query can be answered in  $O(\log n)$  time by Lemma 2. The total running time for case (i) is thus bounded by  $O(n^d \log n)$ .

Case (ii) can be handled in the same way as above.

Case (iii). Finally, consider double-slabs  $D = (h_1, h_2, h_3, h_4)$  enclosing  $P$  of case (iii):  $w(h_1, h_2) = w(h_3, h_4)$  and  $\sum_i |P \cap h_i| \geq d + 2$ . For any four subsets  $Q_1, Q_2, Q_3, Q_4 \subseteq P$ , we define  $D(Q_1, Q_2, Q_3, Q_4) = (h_1, h_2, h_3, h_4)$  to be a minimum-width double-slab such that  $w(h_1, h_2) = w(h_3, h_4)$  and  $P \cap h_i = Q_i$  for  $i = 1, 2, 3, 4$ . Note that  $D(Q_1, Q_2, Q_3, Q_4)$  does not have to enclose  $P$  and it may be undefined when all the conditions cannot be fulfilled at the same time or there is no minimum width even though there are infinitely many such double-slabs. If  $D(Q_1, Q_2, Q_3, Q_4)$  is well defined, then we call it a *candidate double-slab*. Our strategy is to collect all candidate double-slabs over all possible combinations of four subsets and then to test the feasibility of each of them, that is, if it encloses  $P$  by Lemma 2.

It turns out that the case of  $|Q_2| = |Q_3| = 1$  can be handled efficiently.

**Lemma 11.** *Let  $d \geq 2$  be any fixed integer and  $q_2, q_3 \in P$  be two fixed points. The number of candidate double-slabs of the form  $D(Q_1, \{q_2\}, \{q_3\}, Q_4)$  for some  $Q_1, Q_4 \subset P$  is  $O(n^{\lfloor d/2 \rfloor})$ , and all of them can be computed in  $O(n^{\lfloor d/2 \rfloor} + n \log n)$  time. In case of  $d \leq 3$ , given the lower and upper envelopes  $\mathcal{L}$  and  $\mathcal{U}$ , all candidates can be reported in  $O(n)$  time.*

Now, we fix  $2 \leq k \leq d$  and any pair  $(Q_2, Q_3)$  of nonempty subsets of  $P$  with  $|Q_2| + |Q_3| = k$ . We enumerate all candidate double-slabs of the form  $D(Q_1, Q_2, Q_3, Q_4)$  for some  $Q_1, Q_4 \subset P$  in the following way. Compute the subspace  $f = (\text{aff}(Q_2))^\perp \cap (\text{aff}(Q_3))^\perp$  orthogonal to both  $\text{aff}(Q_2)$  and  $\text{aff}(Q_3)$ . As discussed in Sect. 2, we always have  $\dim(f) = d - k + 2$ . As done above for case (i), we project points in  $P$  orthogonally onto  $f$  by projection operator  $\pi_f$ . Observe that all points in  $Q_2$  are projected to a common point  $\bar{q}_2 \in f$  and all points in  $Q_3$  are projected to a common point  $\bar{q}_3 \in f$ . Hence, we can apply Lemma 11 in the  $(d - k + 2)$ -dimensional space  $f$ . This results in  $O(n^{\lfloor (d-k+2)/2 \rfloor})$  candidate double-slabs, taking the same amount of time if  $2 \leq k \leq d - 2$ , or  $O(n \log n)$  time if  $k = d - 1$  or  $d$ . By iterating all  $2 \leq k \leq d$  and all  $\binom{n}{k} \cdot (2^k - 2)$  pairs  $(Q_2, Q_3)$ , we collect a total of  $O(n^{d+1})$  candidate double-slabs. We then test each candidate double-slab if it encloses  $P$  by point location in  $\mathcal{A}$  (Lemma 2) in  $O(\log n)$  time per each.

It remains to consider those double-slabs  $D = (h_1, h_2, h_3, h_4)$  of case (iii) such that  $k = |P \cap h_2| + |P \cap h_3| \geq d + 1$ . In this case, observe that  $(h_2, h_3)$  is an empty slab and its dual segment  $h_2^* h_3^*$  is a candidate stick defined in Sect. 3. (See also Lemma 5.) Thus, we run the algorithm described in Sect. 3.1 to compute all  $O(n^d)$  candidate sticks and, for each candidate sticks  $s$ , we build a corresponding double-slab that encloses  $P$  in  $O(\log n)$  time by ray shooting queries on  $\mathcal{L}$  and  $\mathcal{U}$  at  $u = \pi(s) \in \mathbb{R}^{d-1}$ . So, we can handle this case in  $O(n^d \log n)$  time.

The algorithm described above takes  $O(n^{d+1} \log n)$  time in total to compute an optimal double-slab of case (iii). The most time-consuming part of the algorithm is the case of  $k = d$ : for  $k = d$ , we collect  $O(n)$  candidates for each of  $O(n^d)$

pairs  $(Q_2, Q_3)$  and spend  $O(n^{d+1} \log n)$  time, including the feasibility test. The other cases for  $2 \leq k \leq d-1$  and for  $k \geq d+1$  take only  $O(n^d \log n)$  time in total.

In order to improve the running time, remark that, when  $k = d$ , we reduce the problem in  $\mathbb{R}^d$  to an instance in  $\mathbb{R}^2$ . In the following, we show that these 2-dimensional instances can be handled efficiently in a 3-dimensional instance.

**Lemma 12.** *Let  $d = 3$  and  $q_2, q_3 \in P$  be two fixed points. A minimum-width double-slab  $D = (h_1, h_2, h_3, h_4)$  enclosing  $P$  such that  $w(h_1, h_2) = w(h_3, h_4)$ ,  $q_2 \in h_2$ ,  $q_3 \in h_3$ , and  $|P \cap h_2| + |P \cap h_3| = 3$  can be computed in  $O(n^2)$  time.*

We exploit Lemma 12 as follows: For each pair  $(Q_2, Q_3)$  with  $|Q_2| + |Q_3| = d-1$ , we apply Lemma 12. Then, it also handles the pairs  $(Q'_2, Q_3)$  and  $(Q_2, Q'_3)$  such that  $Q'_2 = Q_2 \cup \{q\}$  and  $Q'_3 = Q_3 \cup \{q\}$  for any point  $q \in P \setminus (Q_2 \cup Q_3)$ . Hence, we can now handle those pairs including  $d$  points in bundles. Since we spend only  $O(n^d \log n)$  time for the cases of  $2 \leq k \leq d-1$ , the final running time is thus improved to  $O(n^{d+1})$ .

**Theorem 4.** *Given a set  $P$  of  $n$  points in  $\mathbb{R}^d$  for any fixed  $d \geq 5$ , a minimum-width double-slab enclosing  $P$  can be computed in  $O(n^{d+1})$  time.*

## References

1. Agarwal, P.K., Har-Peled, S., Varadarajan, K.R.: Approximating extent measures of points. *J. ACM* **51**(4), 606–635 (2004)
2. Agarwal, P.K., Sharir, M.: Efficient randomized algorithms for some geometric optimization problems. *Discr. Comput. Geometry* **16**(4), 317–337 (1996)
3. Agarwal, P.K.: Range searching. In: Goodman, J., O'Rourke, J., Tóth, C. (eds.) *Handbook of Discrete and Computational Geometry*, chap. 40, pp. 1057–1092. CRC Press, 3rd edn. (2018)
4. Agarwal, P.K., Procopiuc, C.M.: Approximation algorithms for projective clustering. *J. Algorithms* **46**(2), 115–139 (2003)
5. Agarwal, P.K., Procopiuc, C.M., Varadarajan, K.R.: Approximation algorithms for a  $k$ -line center. *Algorithmica* **42**(3), 221–230 (2005)
6. Bae, S.W.: Minimum-width double-strip and parallelogram annulus. *Theor. Comput. Sci.* **833**, 133–146 (2020)
7. de Berg, M., Dobrindt, K., Schwarzkopf, O.: On lazy randomized incremental construction. *Discr. Comput. Geometry* **14**, 261–286 (1995)
8. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: *Computational Geometry: Algorithms and Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). <https://doi.org/10.1007/978-3-540-77974-2>
9. Chan, T.M.: Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. *Int. J. Comput. Geom. Appl.* **12**(1–2), 67–85 (2002)
10. Chattopadhyay, S., Das, P.: The  $k$ -dense corridor problems. *Pattern Recogn. Lett.* **11**, 463–469 (1990)
11. Chazelle, B.: Cutting hyperplanes for divide-and-conquer. *Discr. Comput. Geometry* **9**(2), 145–158 (1993)
12. Chazelle, B.: An optimal convex hull algorithm in any fixed dimension. *Discr. Comput. Geometry* **10**, 377–409 (1993)

13. Chazelle, B., Edelsbrunner, H., Guibas, L., Sharir, M.: Diameter, width, closest line pair and parametric searching. *Discr. Comput. Geometry* **10**, 183–196 (1993)
14. Chazelle, B., Friedman, J.: Point location among hyperplanes and unidirectional ray-shooting. *Comput. Geometry: Theor. Appl.* **4**(2), 53–62 (1994)
15. Chung, C., Ahn, T., Bae, S.W., Ahn, H.K.: Parallel line centers with guaranteed separation. In: *Proceedings of the 35th Canadian Conference on Computational Geometry (CCCG 2023)*, pp. 153–160 (2023)
16. Church, R.L., Drezner, Z.: Review of obnoxious facilities location problems. *Comput. Oper. Res.* **138**, 105468 (2022)
17. Das, A.K., Das, S., Mukherjee, J.: Approximation algorithms for orthogonal line centers. *Discret. Appl. Math.* **338**, 69–76 (2023)
18. Díaz-Báñez, J.M., López, M.A., Sellarés, J.A.: Locating an obnoxious plane. *Eur. J. Oper. Res.* **173**(2), 556–564 (2006)
19. Edelsbrunner, H., Seidel, R., Sharir, M.: On the zone theorem for hyperplane arrangements. *SIAM J. Comput.* **22**(2), 418–429 (1993)
20. Glozman, A., Kedem, K., Shpitalnik, G.: On some geometric selection and optimization problems via sorted matrices. *Comput. Geom.: Theory Appl.* **11**(1), 17–28 (1998)
21. Halperin, D., Sharir, M.: Arrangements. In: Goodman, J., O’Rourke, J., Tóth, C. (eds.) *Handbook of Discrete and Computational Geometry*, chap. 28, pp. 723–762. CRC Press, 3rd edn. (2018)
22. Houle, M.E., Maciel, A.: Finding the widest empty corridor through a set of points. *Snapshots of computational and discrete geometry*, pp. 210–213 (1988)
23. Houle, M., Toussaint, G.: Computing the width of a set. In: *Proc. 1st ACM Sympos. Comput. Geom. (SoCG 1985)*, pp. 1–7 (1985)
24. Janardan, R., Preparata, F.P.: Widest-corridor problems. *Nordic J. Comput.* **1**(2), 231–245 (1994)
25. Jaromczyk, J., Kowaluk, M.: The two-line center problem from a polar view: a new algorithm and data structure. In: *Proc. 4th International Workshop Algorithmic Data Structure (WADS 1995)*. *Lecture Notes Comput. Sci.*, vol. 955, pp. 13–25 (1995)
26. Matoušek, J.: Range searching with efficient hierarchical cuttings. *Discr. Comput. Geometry* **10**, 157–182 (1993)
27. Matoušek, J., Schwarzkopf, O.: On ray shooting in convex polytopes. *Discr. Comput. Geometry* **10**, 215–232 (1993)
28. Megiddo, N., Tamir, A.: On the complexity of locating linear facilities in the plane. *Oper. Res. Lett.* **1**(5), 194–197 (1982)
29. Preparata, F.P., Shamos, M.I.: *Computational Geometry*. Springer New York, New York, NY (1985). <https://doi.org/10.1007/978-1-4612-1098-6>
30. Shin, C.S., Shin, S.Y., Chwa, K.Y.: The widest  $k$ -dense corridor problems. *Inf. Process. Lett.* **68**(1), 25–31 (1998)
31. Toussaint, G.: Solving geometric problems with the rotating calipers. In: *Proceedings of the IEEE MELECON (1983)*