

Maximum Volume Subset Selection for Anchored Boxes

Karl Bringmann*

Sergio Cabello†

Michael T.M. Emmerich‡

Abstract

Let B be a set of n axis-parallel boxes in \mathbb{R}^d such that each box has a corner at the origin and the other corner in the positive quadrant of \mathbb{R}^d , and let k be a positive integer. We study the problem of selecting k boxes in B that maximize the volume of the union of the selected boxes. This research is motivated by applications in skyline queries for databases and in multicriteria optimization, where the problem is known as the *hypervolume subset selection problem*. It is known that the problem can be solved in polynomial time in the plane, while the best known running time in any dimension $d \geq 3$ is $\Omega\left(\binom{n}{k}\right)$. We show that:

- The problem is NP-hard already in 3 dimensions.
- In 3 dimensions, we break the bound $\Omega\left(\binom{n}{k}\right)$, by providing an $n^{O(\sqrt{k})}$ algorithm.
- For any constant dimension d , we present an efficient polynomial-time approximation scheme.

*Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany.

†Department of Mathematics, IMFM, and Department of Mathematics, FMF, University of Ljubljana, Slovenia. Supported by the Slovenian Research Agency, program P1-0297 and project L7-5459.

‡Leiden Institute of Advanced Computer Science (LIACS), Leiden University, the Netherlands.

1 Introduction

An *anchored box* is an orthogonal range of the form $\text{BOX}(p) := [0, p_1] \times \dots \times [0, p_d] \subset \mathbb{R}_{\geq 0}^d$, spanned by the point $p \in \mathbb{R}_{> 0}^d$. This paper is concerned with the problem **VOLUME SELECTION**: Given a set P of n points in $\mathbb{R}_{> 0}^d$, select k points in P maximizing the volume of the union of their anchored boxes. That is, we want to compute

$$\text{VOLSEL}(P, k) := \max_{S \subseteq P, |S|=k} \text{VOL}\left(\bigcup_{p \in S} \text{BOX}(p)\right),$$

as well as a set $S^* \subseteq P$ of size k realizing this value. Here, VOL denotes the usual volume.

Motivation This geometric problem is of key importance in the context of multicriteria optimization and decision analysis, where it is known as the *hypervolume subset selection problem (HSSP)* [2–4, 12, 13, 24]. In this context, the points in P correspond to solutions of an optimization problem with d objectives, and the goal is to find a small subset of P that “represents” the set P well. The quality of a representative subset $S \subseteq P$ is measured by the volume of the union of the anchored boxes spanned by points in S ; this is also known as the *hypervolume indicator* [36]. Note that with this quality indicator, finding the optimal size- k representation is equivalent to our problem $\text{VOLSEL}(P, k)$. In applications, such bounded-size representations are required in archivers for non-dominated sets [23] and for multicriteria optimization algorithms and heuristics [3, 7, 10].¹ Besides, the problem has recently received attention in the context of skyline operators in databases [17].

In 2 dimensions, the problem can be solved in polynomial time [2, 13, 24], which is used in applications such as analyzing benchmark functions [2] and efficient postprocessing of multiobjective algorithms [12]. A natural question is whether efficient algorithms also exist in dimension $d \geq 3$, and thus whether these applications can be pushed beyond two objectives.

In this paper, we answer this question negatively, by proving that **VOLUME SELECTION** is NP-hard already in 3 dimensions. We then consider the question whether the previous $\Omega\binom{n}{k}$ bound can be improved, which we answer affirmatively in 3 dimensions. Finally, for any constant dimension, we improve the best-known $(1 - 1/e)$ -approximation to an efficient polynomial-time approximation scheme (EPTAS). See Section 1.2 for details.

1.1 Further Related Work

Klee’s Measure Problem To compute the volume of the union of n (not necessarily anchored) axis-aligned boxes in \mathbb{R}^d is known as Klee’s measure problem. The fastest known algorithm takes time² $O(n^{d/2})$, which can be improved to $O(n^{d/3} \text{polylog}(n))$ if all boxes are cubes [15]. By a simple reduction [8], the same running time as on cubes can be obtained on anchored boxes, which can be improved to $O(n \log n)$ for $d \leq 3$ [6]. These results are relevant to this paper because Klee’s measure problem on anchored boxes (spanned by the points in P) is a special case of **VOLUME SELECTION** (by calling $\text{VOLSEL}(P, |P|)$).

Chan [14] gave a reduction from k -Clique to Klee’s measure problem in $2k$ dimensions. This proves NP-hardness of Klee’s measure problem when d is part of the input (and thus d can be as large as n). Moreover, since k -Clique has no $f(k) \cdot n^{o(k)}$ -time algorithm under the Exponential Time Hypothesis [16], Klee’s measure problem has no $f(d) \cdot n^{o(d)}$ -time algorithm under the same assumption. The same hardness results also hold for Klee’s measure problem on anchored boxes, by a reduction in [8] (NP-hardness was first proven in [11]).

Finally, we mention that Klee’s measure problem has a very efficient randomized $(1 \pm \varepsilon)$ -approximation algorithm in time $O(n \log(1/\delta)/\varepsilon^2)$ with error probability δ [9].

¹We remark that in these applications the anchor point is often not the origin, however, by a simple translation we can move our anchor point from $(0, \dots, 0)$ to any other point in \mathbb{R}^d .

²In O -notation, we always assume d to be a constant, and $\log(x)$ is to be understood as $\max\{1, \log(x)\}$.

Known Results for Volume Selection As mentioned above, 2-dimensional VOLUME SELECTION can be solved in polynomial time; the initial $O(kn^2)$ algorithm [2] was later improved to $O((n-k)k + n \log n)$ [13, 24]. In higher dimensions, by enumerating all size- k subsets and solving an instance of Klee’s measure problem on anchored boxes for each one, there is an $O(\binom{n}{k} k^{d/3} \text{polylog}(k))$ algorithm. For small $n-k$, this can be improved to $O(n^{d/2} \log n + n^{n-k})$ [10]. VOLUME SELECTION is NP-hard when d is part of the input, since the same holds already for Klee’s measure problem on anchored boxes. However, this does not explain the exponential dependence on k for constant d .

Since the volume of the union of boxes is a submodular function (see, e.g., [33]), the greedy algorithm for submodular function maximization [28] yields a $(1 - 1/e)$ -approximation of $\text{VOLSEL}(P, k)$. This algorithm solves $O(nk)$ instances of Klee’s measure problem on at most k anchored boxes, and thus runs in time $O(nk^{d/3+1} \text{polylog}(k))$. Using [9], this running time improves to $O(nk^2 \log(1/\delta)/\varepsilon^2)$, at the cost of decreasing the approximation ratio to $1 - 1/e - \varepsilon$ and introducing an error probability δ . See [20] for related results in 3 dimensions.

A problem closely related to VOLUME SELECTION is CONVEX HULL SUBSET SELECTION: Given n points in \mathbb{R}^d , select k points that maximize the volume of their convex hull. For this problem, NP-hardness was recently announced in the case $d = 3$ [30].

1.2 Our Results

In this paper we push forward the understanding of VOLUME SELECTION. We prove that VOLUME SELECTION is NP-hard already for $d = 3$ (Section 3). Previously, NP-hardness was only known when d is part of the input and thus can be as large as n . Moreover, this establishes VOLUME SELECTION as another example for problems that can be solved in polynomial time in the plane but are NP-hard in three or more dimensions (see also [5, 26]).

In the remainder, we focus on the regime where $d \geq 3$ is a constant and $k \ll n$. All known algorithms (explicitly or implicitly) enumerate all size- k subsets of the input set P and thus take time $\Omega(\binom{n}{k}) = n^{\Omega(k)}$. In 3 dimensions, we break this time bound by providing an $n^{O(\sqrt{k})}$ algorithm (Section 4). To this end, we project the 3-dimensional VOLUME SELECTION to a 2-dimensional problem and then use planar separator techniques.

Finally, in Section 5 we design an EPTAS for VOLUME SELECTION. More precisely, we present a $(1 - \varepsilon)$ -approximation algorithm running in time $O(n \cdot \varepsilon^{-d} (\log n + k + 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d}))$, for any constant dimension d . Note that the “combinatorial explosion” is restricted to d and ε ; for any constant d, ε the algorithm runs in time $O(n(k + \log n))$. This improves the previously best-known $(1 - 1/e)$ -approximation, even in terms of running time.

2 Preliminaries

All boxes considered in the paper are axis-parallel and anchored at the origin. For points $p = (p_1, \dots, p_d), q = (q_1, \dots, q_d) \in \mathbb{R}^d$, we say that p *dominates* q if $p_i \geq q_i$ for all $1 \leq i \leq d$. For $p = (p_1, \dots, p_d) \in \mathbb{R}_{>0}^d$, we let $\text{BOX}(p) := [0, p_1] \times \dots \times [0, p_d]$. Note that $\text{BOX}(p)$ is the set of all points $q \in \mathbb{R}_{>0}^d$ that are dominated by p . A *point set* P is a set of points in $\mathbb{R}_{>0}^d$. We denote the union $\bigcup_{p \in P} \text{BOX}(p)$ by $\mathcal{U}(P)$. The usual Euclidean volume is denoted by VOL . With this notation, we set

$$\mu(P) := \text{VOL}(\mathcal{U}(P)) = \text{VOL}\left(\bigcup_{p \in P} \text{BOX}(p)\right) = \text{VOL}\left(\bigcup_{p \in P} [0, p_1] \times \dots \times [0, p_d]\right).$$

We study VOLUME SELECTION: Given a point set P of size n and $0 \leq k \leq n$, compute

$$\text{VOLSEL}(P, k) := \max_{S \subseteq P, |S|=k} \mu(S).$$

Note that we can relax the requirement $|S| = k$ to $|S| \leq k$ without changing this value.

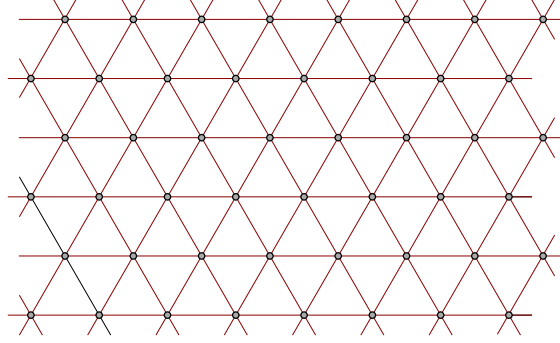


Figure 1: Triangular grid Γ .

3 Hardness in 3 Dimensions

We consider the following decision variant of 3-dimensional VOLUME SELECTION.

3D VOLUME SELECTION

Input: A triple (P, k, V) , where P is a set of points in $\mathbb{R}_{>0}^3$, k is a positive integer and V is a positive real value.

Question: Is there a subset $Q \subseteq P$ of k points such that $\mu(Q) \geq V$?

We are going to show that the problem is NP-complete. First, we show that an intermediate problem about selecting a large independent set in a given induced subgraph of the triangular grid is NP-hard. The reduction for this problem is from independent set in planar graphs of maximum degree 3. Then we argue that this problem can be embedded using boxes whose points lie in two parallel planes. One plane is used to define the triangular-grid-like structure and the other is used to encode the subset of vertices that describe the induced subgraph of the grid.

3.1 Triangular Grid

Let Γ be the infinite graph with vertex set and edge set (see Figure 1)

$$V(\Gamma) = \{(i + j \cdot 1/2, j \cdot \sqrt{3}/2) \mid i, j \in \mathbb{N}\},$$

$$E(\Gamma) = \{ab \mid a, b \in V(\Gamma), \text{ the Euclidean distance between } a \text{ and } b \text{ is exactly } 1\}.$$

First we show that the following intermediate problem, which is closely related to independent set, is NP-hard.

INDEPENDENT SET ON INDUCED TRIANGULAR GRID

Input: A pair (A, ℓ) , where A is a subset of $V(\Gamma)$ and ℓ is a positive integer.

Question: Is there a subset $B \subseteq A$ of size ℓ such that no two vertices in B are connected by an edge of $E(\Gamma)$?

Lemma 3.1. INDEPENDENT SET ON INDUCED TRIANGULAR GRID is NP-complete.

Proof. It is obvious that the problem is in NP.

Garey and Johnson [19] show that the problem VERTEX COVER is NP-complete for planar graphs of degree at most 3. Since a subset $U \subseteq V(G)$ is a vertex cover of graph G if and only if $V(G) \setminus U$ is an independent set of G , it follows that the problem INDEPENDENT SET is NP-complete for planar graphs of degree at most 3. For the rest of the proof, let G be a planar graph of degree at most 3.

Let us define a Γ -*representation* of G to be a pair (H, φ) , where $H \subset \Gamma$ and φ is a mapping, with the following properties:

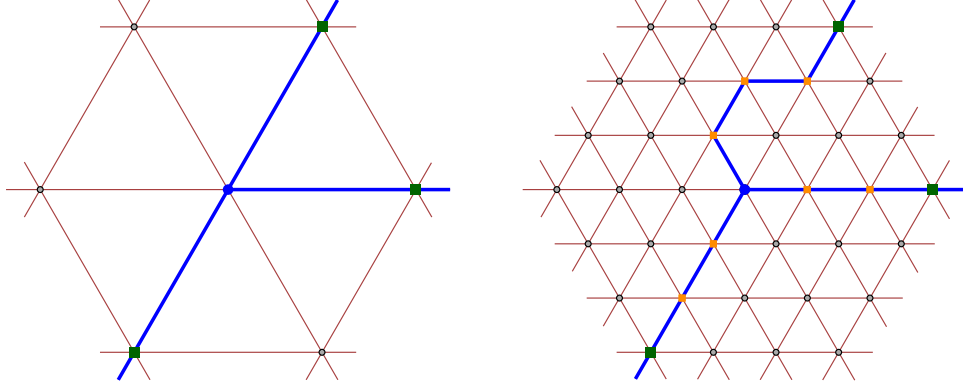


Figure 2: Transformation to get an induced subgraph of the triangular grid. Vertices from the subdivision of edges are green squares.

- Each vertex u of G is mapped to a distinct vertex $\varphi(u)$ of H .
- Each edge uv of G is mapped to a simple path $\varphi(uv)$ contained in H and connecting $\varphi(u)$ to $\varphi(v)$.
- For each two distinct edges uv and $u'v'$ of G , the paths $\varphi(uv)$ and $\varphi(u'v')$ are disjoint except at the common endpoints $\{\varphi(u), \varphi(v)\} \cap \{\varphi(u'), \varphi(v')\}$.
- The graph H is precisely the union of $\varphi(u)$ and $\varphi(uv)$ over all vertices u and edges uv of G .

Note that if (H, φ) is a Γ -representation of G then H is a subdivision of G . The map ϕ identifies which parts of H correspond to which parts of G .

A planar graph G with n vertices and maximum degree 3 (and also 4) can be drawn in a square grid of polynomial size, and such a drawing can be obtained in polynomial time, see, e.g., the results by Storer [31] or by Tamassia and Tollis [32]. Applying the shear mapping $(x, y) \mapsto (x + y/2, y\sqrt{2}/3)$ to the plane, the square grid becomes a subgraph of Γ . Therefore, we can obtain a Γ -representation (H_1, φ_1) of G of polynomial size. Note that we only use edges of Γ that are horizontal or have positive slope; edges of Γ with negative slope are not used.

Next, we obtain another Γ -representation (H_2, φ_2) such that H_2 is an *induced* subgraph of Γ . Induced means that two vertices of H_2 are connected with an edge in H_2 if and only if the edge exists in Γ . For this, we first scale up the Γ -representation (H_1, φ_1) by a factor 2 so that each edge of H_1 becomes a 2-edge path. The new vertices used in the subdivision have degree 2 and its 2 incident edges have the same orientation. After the subdivision, vertices of degree 3 look like in Figure 2. Scaling up the figure by a factor of 3, and rerouting within a small neighbourhood of each vertex v that was already in H_1 , we obtain a Γ -representation (H_2, φ_2) such that H_2 is an *induced* subgraph of Γ . See Figure 2 for an example of such a local transformation.

Now we have a Γ -representation (H_2, φ_2) such that H_2 is an induced subgraph of Γ . We want to obtain another Γ -representation where for each edge $uv \in E(G)$ the path $\varphi_2(uv)$ uses an even number of interior edges. For this, we can slightly reroute each path $\varphi_2(uv)$ that has an odd number of interior points, see Figure 3. To make sure that the graph is still induced, we can first scale up the situation by a factor 2, and then reroute all the edges $\varphi_2(uv)$ that use an odd number of interior vertices. (This is actually all the edges $uv \in E(G)$ because of the scaling.) Let (H_3, φ_3) be the resulting Γ -representation of G . Note that H_3 is an induced subgraph of Γ and it is a subdivision of G where each edge is subdivided an even number of times.

Let $\alpha(G)$ denote the size of the largest independent set in G . For each edge uv of G , let $2k_{uv}$ be the number of internal vertices in the path $\varphi_3(uv)$. Then $\alpha(H_3) = \alpha(G) + \sum_{uv \in E(G)} k_{uv}$.

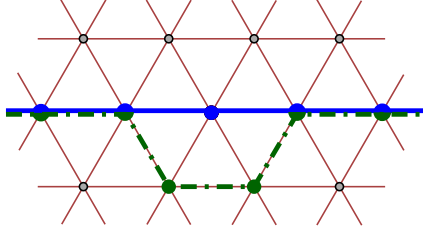


Figure 3: Choosing the parity of paths.

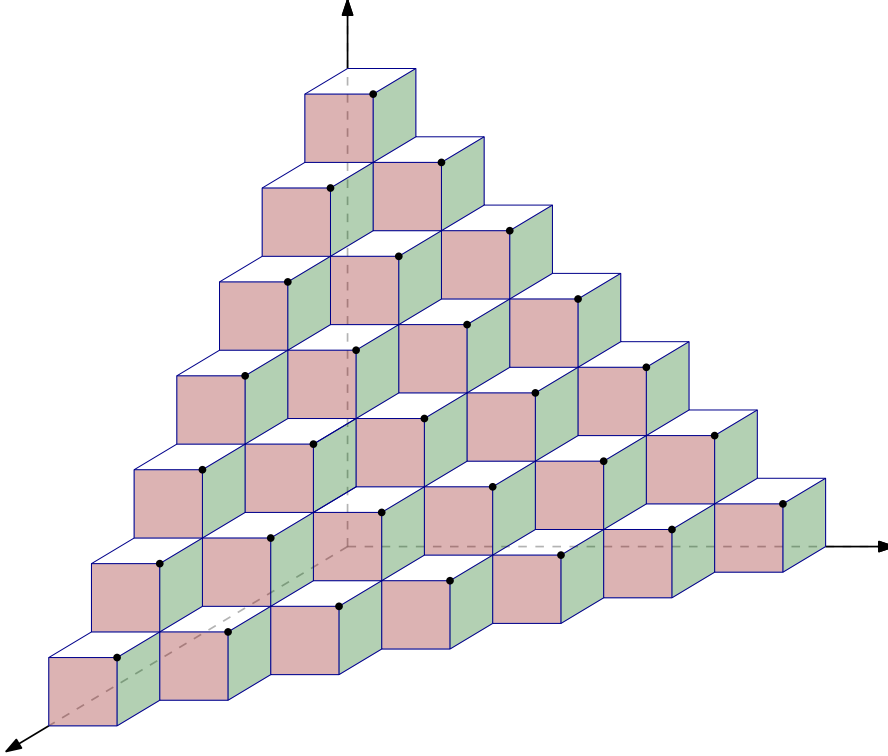


Figure 4: The point set P_m and the boxes $\text{BOX}(p)$, with $p \in P_m$, for $m = 9$.

Indeed, we can obtain H_3 from G by repeatedly replacing an edge by a 3-edge path, i.e., making 2 subdivisions on the same edge. Moreover, any such replacement increases the size of the largest independent set by exactly 1.

It follows that the problem INDEPENDENT SET is NP-complete in *induced* subgraphs of the triangular grid Γ . This is precisely the problem INDEPENDENT SET ON INDUCED TRIANGULAR GRID, where we take A to be the set of vertices defining the induced subgraph. \square

3.2 The Point Set

Let $m \geq 3$ be an arbitrary integer and consider the point set P_m defined by (see Figure 4)

$$P_m = \{(x, y, z) \in \mathbb{N}^3 \mid x + y + z = m\}.$$

Standard induction shows that the set P_m has $1 + 2 + \dots + (m - 2) = (m - 1)(m - 2)/2$ points and that

$$\mu(P_m) = \text{VOL} \left(\bigcup_{p \in P_m} \text{BOX}(p) \right) = m(m - 1)(m - 2)/6.$$

This last number appears as sequence A000292, tetrahedral (or triangular pyramidal) numbers, in [27].

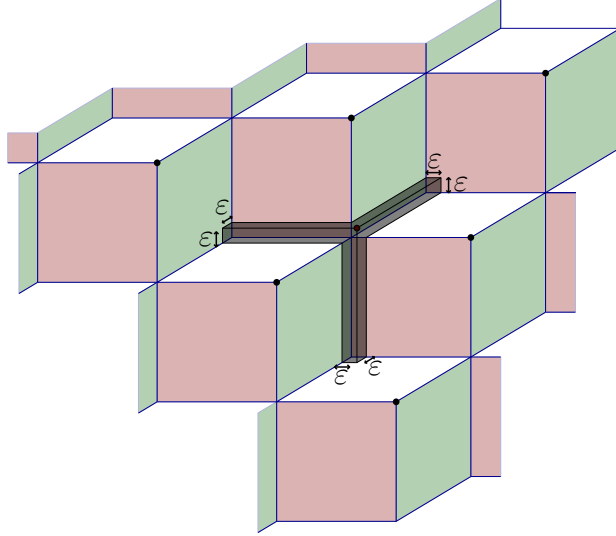


Figure 5: The point $q = p + \Delta_\varepsilon$ and the set $\text{DIFF}(q)$.

Consider the real number $\varepsilon = 1/4m^2$, and define the vector $\Delta_\varepsilon = (\varepsilon, \varepsilon, \varepsilon)$. Note that ε is much smaller than 1. For each point $p \in P_{m-1}$, consider the point $p + \Delta_\varepsilon$, see Figure 5. Let us define the set Q_m to be

$$Q_m = \{p + \Delta_\varepsilon \mid p \in P_{m-1}\}.$$

It is clear that Q_m has $|P_{m-1}| = (m-2)(m-3)/2$ points, for $m \geq 3$. The points of Q_m lie on the plane $x + y + z = m - 1 + 3\varepsilon$.

For each point q of Q_m define

$$\text{DIFF}(q) = \mathcal{U}(P_m \cup \{q\}) \setminus \mathcal{U}(P_m) = \left(\bigcup_{p \in P_m \cup \{q\}} \text{BOX}(p) \right) \setminus \left(\bigcup_{p \in P_m} \text{BOX}(p) \right).$$

Note that $\text{DIFF}(q)$ is the union of 3 boxes of size $\varepsilon \times \varepsilon \times 1$ and a cube of size $\varepsilon \times \varepsilon \times \varepsilon$, see Figure 5. To get the intuition for the following lemma, see Figure 6.

Lemma 3.2. *Consider any $Q' \subseteq Q_m$.*

- *If the sets $\text{DIFF}(q)$, for all $q \in Q'$, are pairwise disjoint, then $\mu(P_m \cup Q') = \mu(P_m) + |Q'| \cdot (3\varepsilon^2 + \varepsilon^3)$.*
- *If Q' contains two points q_0 and q_1 such that $\text{DIFF}(q_0)$ and $\text{DIFF}(q_1)$ intersect, then $\mu(P_m \cup Q') < \mu(P_m) + |Q'| \cdot (3\varepsilon^2 + \varepsilon^3)$.*

Proof. Note that for each $q \in Q_m$ we have

$$\mu(P_m \cup \{q\}) - \mu(P_m) = \text{VOL}(\text{DIFF}(q)) = 3\varepsilon^2 + \varepsilon^3.$$

If the sets $\{\text{DIFF}(q) \mid q \in Q'\}$ are pairwise disjoint then

$$\begin{aligned} \mu(P_m \cup Q') &= \mu(P_m) + \text{VOL} \left(\bigcup_{q \in Q'} \text{DIFF}(q) \right) \\ &= \mu(P_m) + \sum_{q \in Q'} \text{VOL}(\text{DIFF}(q)) \\ &= \mu(P_m) + |Q'| \cdot (3\varepsilon^2 + \varepsilon^3). \end{aligned}$$

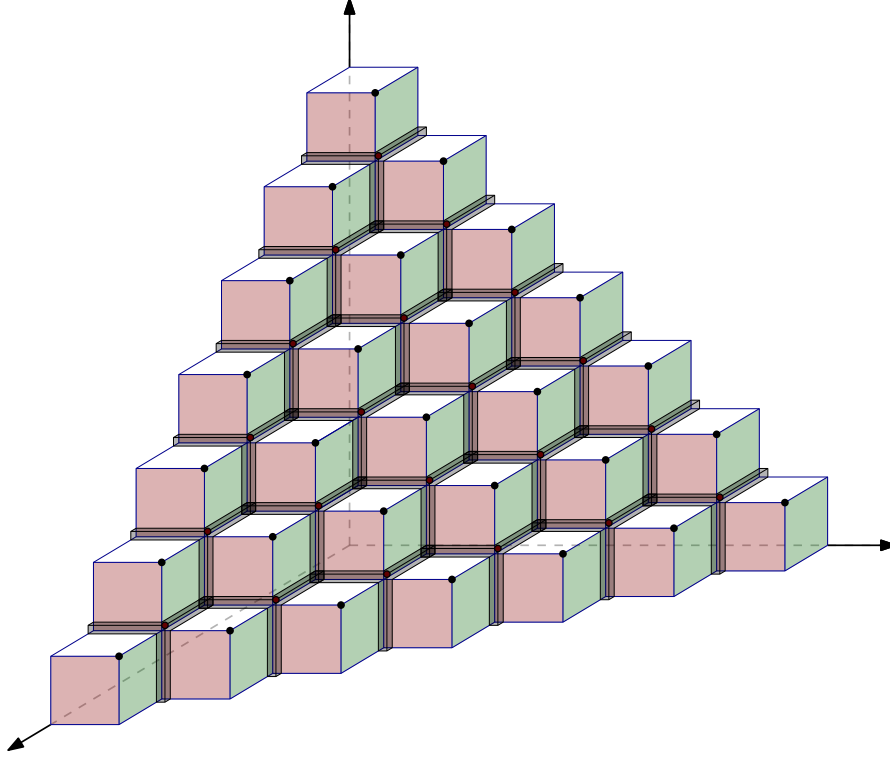


Figure 6: The sets $\text{DIFF}(q)$ for all $q \in Q_m$.

Consider now the case when Q' contains two points q_0 and q_1 such that $\text{DIFF}(q_0)$ and $\text{DIFF}(q_1)$ intersect. The geometry of the point set Q' implies that $\text{DIFF}(q_0)$ and $\text{DIFF}(q_1)$ intersect in a cube of size $\varepsilon \times \varepsilon \times \varepsilon$, see Figure 6. Therefore, we have

$$\begin{aligned}
\mu(P_m \cup Q') &= \mu(P_m) + \text{VOL} \left(\bigcup_{q \in Q'} \text{DIFF}(q) \right) \\
&\leq \mu(P_m) + \sum_{q \in Q'} \text{VOL}(\text{DIFF}(q)) - \text{VOL}(\text{DIFF}(q_0) \cap \text{DIFF}(q_1)) \\
&= \mu(P_m) + |Q'| \cdot (3\varepsilon^2 + \varepsilon^3) - \varepsilon^2 \\
&< \mu(P_m) + |Q'| \cdot (3\varepsilon^2 + \varepsilon^3). \quad \square
\end{aligned}$$

We can define naturally a graph T_m on the set Q_m by using the intersection of the sets $\text{DIFF}(\cdot)$. The vertex set of T_m is Q_m , and two points $q, q' \in Q_m$ define an edge qq' of T_m if and only if $\text{DIFF}(q)$ and $\text{DIFF}(q')$ intersect, see Figure 7. Simple geometry shows that T_m is isomorphic to a part of the triangular grid Γ . Thus, choosing m large enough, we can get an arbitrarily large portion of the triangular grid Γ . Note that a subset of vertices $Q' \subseteq Q_m$ is independent in T_m if and only if the sets $\{\text{DIFF}(q) \mid q \in Q'\}$ are pairwise disjoint.

We next show that picking points in P_m has higher priority than picking points in Q_m .

Lemma 3.3. *Let P' be a subset of P_m such that $P_m \setminus P'$ is not empty. Then $\mu(P' \cup Q_m) < \mu(P_m)$.*

Proof. Assume that $P_m \setminus P'$ contains exactly one point, denoted by p . Having a smaller set P' can only decrease the value of $\mu(P' \cup Q_m)$. Then

$$\mu(P') = \mu(P_m) - 1.$$

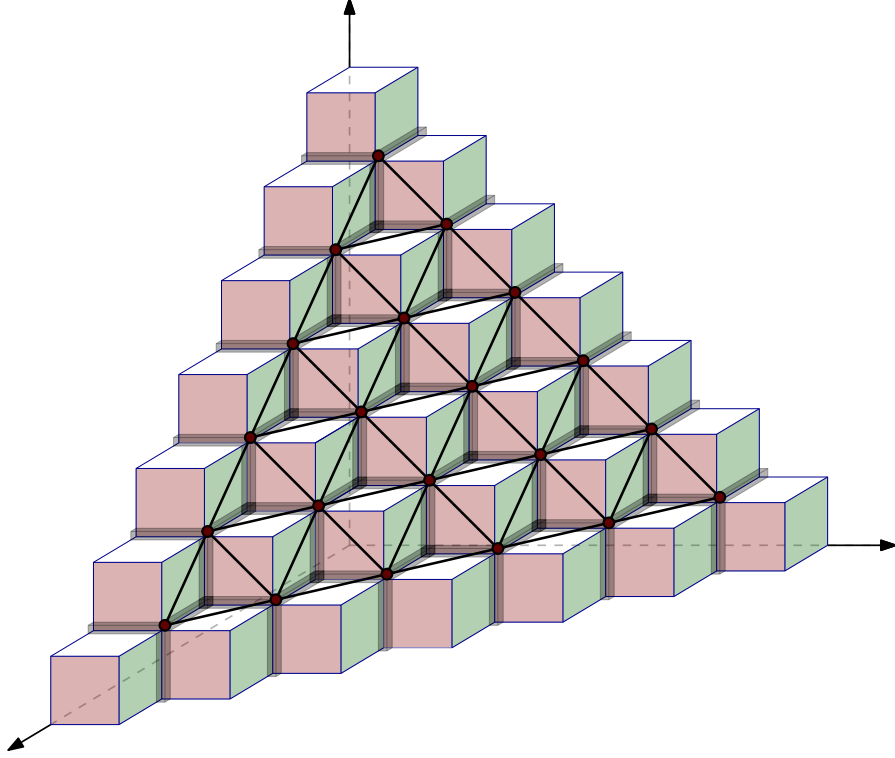


Figure 7: The graph T_m for $m = 9$.

Consider the sets of 3 points

$$Q_m^1(p) = \{(p_x - 1, p_y, p_z) + \Delta_\varepsilon, (p_x, p_y - 1, p_z) + \Delta_\varepsilon, (p_x, p_y, p_z - 1) + \Delta_\varepsilon\} \subseteq Q_m,$$

$$Q_m^2(p) = \{(p_x - 1, p_y - 1, p_z + 1) + \Delta_\varepsilon, (p_x + 1, p_y - 1, p_z - 1) + \Delta_\varepsilon,$$

$$(p_x - 1, p_y + 1, p_z - 1) + \Delta_\varepsilon\} \subseteq Q_m.$$

Figure 8 is useful for the following computations. For each point $q \in Q_m^1(p)$ we have

$$\mu(P' \cup q) = \mu(P') + \text{VOL}(\text{DIFF}(q)) + \varepsilon.$$

For each point $q \in Q_m^2(p)$ we have

$$\mu(P' \cup q) = \mu(P') + \text{VOL}(\text{DIFF}(q)) + \varepsilon^2.$$

Using that $\varepsilon^2 \leq \varepsilon$ because $0 < \varepsilon < 1$, we get

$$\forall q \in Q_m^1(p) \cup Q_m^2(p) : \mu(P' \cup q) \leq \mu(P') + \text{VOL}(\text{DIFF}(q)) + \varepsilon.$$

For all points q of $Q_m \setminus (Q_m^1(p) \cup Q_m^2(p))$ we have

$$\mu(P' \cup q) = \mu(P') + \text{VOL}(\text{DIFF}(q)).$$

We thus have

$$\begin{aligned} \mu(P' \cup Q_m) &\leq \mu(P') + \sum_{q \in Q_m} \text{VOL}(\text{DIFF}(q)) + \sum_{q \in Q_m^1(p) \cup Q_m^2(p)} \varepsilon \\ &= \mu(P_m) - 1 + |Q_m| \cdot (3\varepsilon^2 + \varepsilon^3) + 6 \cdot \varepsilon \\ &\leq \mu(P_m) - 1 + \frac{(m-2)(m-3)}{2} \cdot 4 \cdot \varepsilon + 6 \cdot \varepsilon \\ &< \mu(P_m), \end{aligned}$$

where the last step uses $\varepsilon = 1/4m^2$.

□

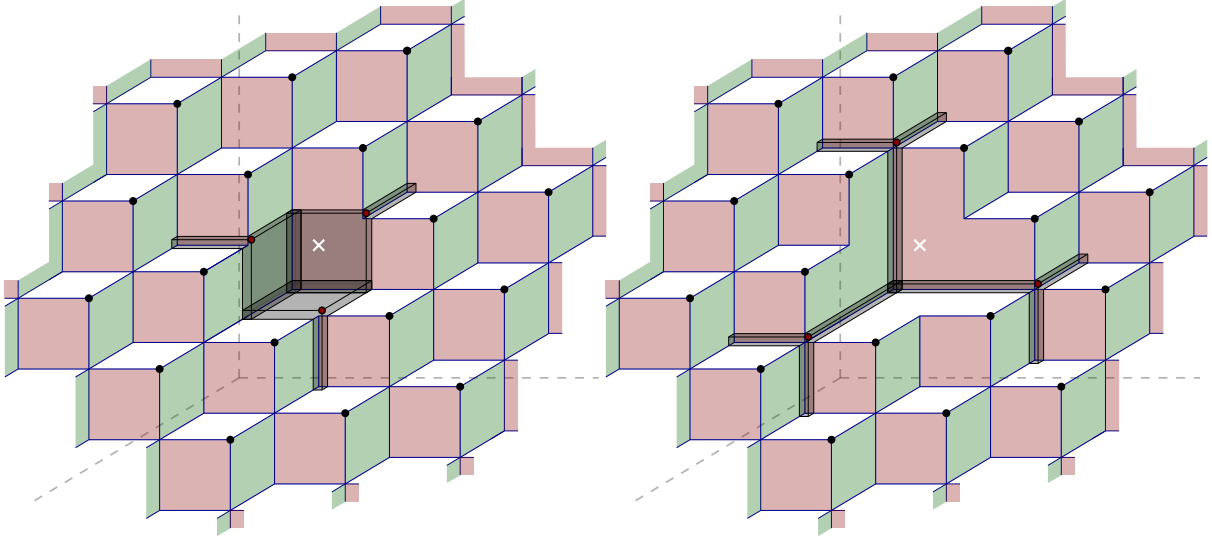


Figure 8: Image for the proof of Lemma 3.3. The point p of P_m that is missing in P' is indicated with a white cross. Left: the contribution of points from $Q_m^1(p)$. Right: the contribution of points from $Q_m^2(p)$.

3.3 The Reduction

We are now ready to prove NP-completeness of 3D VOLUME SELECTION.

Theorem 3.4. *The problem 3D VOLUME SELECTION is NP-complete.*

Proof. It is obvious that the problem is in NP. To show hardness we reduce from the problem INDEPENDENT SET INDUCED TRIANGULAR GRID, shown to be NP-complete in Lemma 3.1.

Consider an instance (A, ℓ) to INDEPENDENT SET ON INDUCED TRIANGULAR GRID, where A is a subset of the vertices of the triangular grid Γ and ℓ is an integer. Take m large enough so that T_m is isomorphic to an induced subgraph of Γ that contains A . Recall that $\varepsilon = 1/4m^2$. For each vertex v of T_m let $\psi_\Gamma(v)$ be the corresponding vertex of Γ . For each subset B of A , let $Q_m(B)$ be the subset of T_m that corresponds to B , that is, $Q_m(B) = \{q \in Q_m \mid \psi_\Gamma(q) \in B\}$.

Consider the set of points $P = P_m \cup Q_m(A)$, the parameter $k = (m-1)(m-2)/2 + \ell$, and the value $V = \frac{m(m-1)(m-2)}{6} + \ell \cdot (3\varepsilon^2 + \varepsilon^3)$. We claim that (A, ℓ) is a yes-instance for INDEPENDENT SET ON INDUCED TRIANGULAR GRID if and only if (P, k, V) is a yes-instance for 3D VOLUME SELECTION.

If (A, ℓ) is a yes-instance for INDEPENDENT SET ON INDUCED TRIANGULAR GRID, there is a subset $B \subseteq A$ of ℓ independent vertices in Γ . This implies that $Q_m(B)$ is an independent set in T_m , that is, the sets $\{\text{DIFF}(q) \mid q \in Q_m(B)\}$ are pairwise disjoint. Lemma 3.2 then implies that

$$\begin{aligned} \mu(P_m \cup Q_m(B)) &= \mu(P_m) + |B| \cdot (3\varepsilon^2 + \varepsilon^3) \\ &= \frac{m(m-1)(m-2)}{6} + \ell \cdot (3\varepsilon^2 + \varepsilon^3) \\ &= V. \end{aligned}$$

Therefore $P_m \cup Q_m(B)$ is a subset of P with $|P_m| + |B| = (m-1)(m-2)/2 + \ell = k$ points such that $\mu(P_m \cup Q_m(B)) = V$. It follows that (P, k, V) is a yes-instance for 3D VOLUME SELECTION.

Assume now that (P, k, V) is a yes-instance for 3D VOLUME SELECTION. This means that P contains a subset Q of k points such that

$$\mu(Q) \geq V = \frac{m(m-1)(m-2)}{6} + \ell \cdot (3\varepsilon^2 + \varepsilon^3) = \mu(P_m) + \ell \cdot (3\varepsilon^2 + \varepsilon^3) > \mu(P_m).$$

Because of Lemma 3.3, it must be that P_m is contained in Q , as otherwise we would have $\mu(Q) < \mu(P_m)$. Since we have $P_m \subset Q$ and $P = P_m \cup Q_m(A)$, we obtain that Q is $P_m \cup Q_m(B)$ for some $B \subseteq A$. Moreover, $|B| = k - |P_m| = \ell$. By Lemma 3.2, if $Q_m(B)$ is not an independent set in T_m , we have

$$\mu(Q) = \mu(P_m \cup Q_m(B)) < \mu(P_m) + \ell(3\varepsilon^2 + \varepsilon) = V,$$

which contradicts the assumption that $\mu(Q) \geq V$. Therefore it must be that $Q_m(B)$ is an independent set in T_m . It follows that $B \subset A$ has size ℓ and forms an independent set in Γ , and thus (A, ℓ) is a yes-instance for INDEPENDENT SET ON INDUCED TRIANGULAR GRID. \square

4 Exact Algorithm in 3 Dimensions

In this section we design an algorithm to solve VOLUME SELECTION in 3 dimensions in time $n^{O(\sqrt{k})}$. The main insight is that, for an optimal solution Q^* , the boundary of $\mathcal{U}(Q^*)$ is a planar graph with $O(k)$ vertices, and therefore has a balanced separator with $O(\sqrt{k})$ vertices. We would like to guess the separator, break the problem into two subproblems, and solve each of them recursively. This basic idea leads to a few technical challenges to take care of. One obstacle is that subproblems should be really independent because we do not want to double count some covered parts. Essentially, a separator in the graph-theory sense does not imply independent subproblems in our context. Another technicality is that some of the subproblems that we encounter recursively cannot be solved optimally; we can only get a lower bound to the optimal value. However, for the subproblems that define the optimal solution at the higher level of the recursion, we do compute an optimal solution.

Let P be a set of n points in the positive quadrant of \mathbb{R}^3 . Through our discussion, we will assume that P is fixed and thus drop the dependency on P and n from the notation. We can assume that no point of P is dominated by another point of P . Using an infinitesimal perturbation of the points, we can assume that all points have all coordinates different. Indeed, we can replace each point p by the point $p + i(\varepsilon, \varepsilon, \varepsilon)$, where i is a different integer for each point of P and $\varepsilon > 0$ is an infinitesimal value or a value that is small enough.

Let M be the largest x - or y -coordinate in P , thus $M = \max\{p_x, p_y \mid p \in P\}$. We define σ to be the square in \mathbb{R}^2 defined by $[-1, M + 1] \times [-1, M + 1]$. It has side length $M + 2$.

For each subset Q of P , consider the projection of $\mathcal{U}(Q)$ onto the xy -plane. This defines a plane graph, which we denote by $G(Q)$, and which we define precisely in the following, see Figure 9. We consider $G(Q)$ as a geometric, embedded graph where each vertex is a point and each edge is (drawn as) a straight-line segment, in fact, a horizontal or vertical straight-line segment on the xy -plane. There are different types of vertices in $G(Q)$. The projection of each point $q \in Q$ defines a vertex, which we denote by v_q . When for two distinct points $q, q' \in Q$ the boundary of the projection of the boxes $\text{BOX}(q)$ and the boundary of the projection of $\text{BOX}(q')$ intersect outside the x - and y -axis, then they do so exactly once because of our assumption on general position, and this defines a vertex that we denote by $v_{q,q'}$. (Not all pairs (q, q') define such a vertex.) Additionally, each point $q \in Q$ defines a vertex $v_{x,q}$ at position $(q_x, 0)$ and a vertex $v_{y,q}$ at position $(0, q_y)$. Finally, we have a vertex $v_{x,y}$ placed at the origin. The vertices of $G(Q)$ are connected in a natural way: the boundary of the visible part of $\text{BOX}(q)$ connects the points that appear on that boundary. In particular, the vertices on the x -axis are connected and so do those on the y -axis. Since we assume general position, each vertex uniquely determines the boxes that define it. Each vertex $q \in Q$ defines a bounded face $f(q, Q)$ in $G(Q)$. This is the projection of the face on the boundary of $\mathcal{U}(Q)$ contained in the plane $\{(x, y, z) \in \mathbb{R}^3 \mid z = q_z\}$, see Figure 9, right. In fact, each bounded face of $G(Q)$ is $f(q, Q)$ for some $q \in Q$.

We triangulate each bounded face $f(q, Q)$ of $G(Q)$ *canonically*, as follows, see Figure 10. The boundary of a bounded face $f(q, Q)$ is made of a top horizontal segment $t(q, Q)$ (which may contain several edges of the graph), a right vertical segment $r(q, Q)$ (which may contain

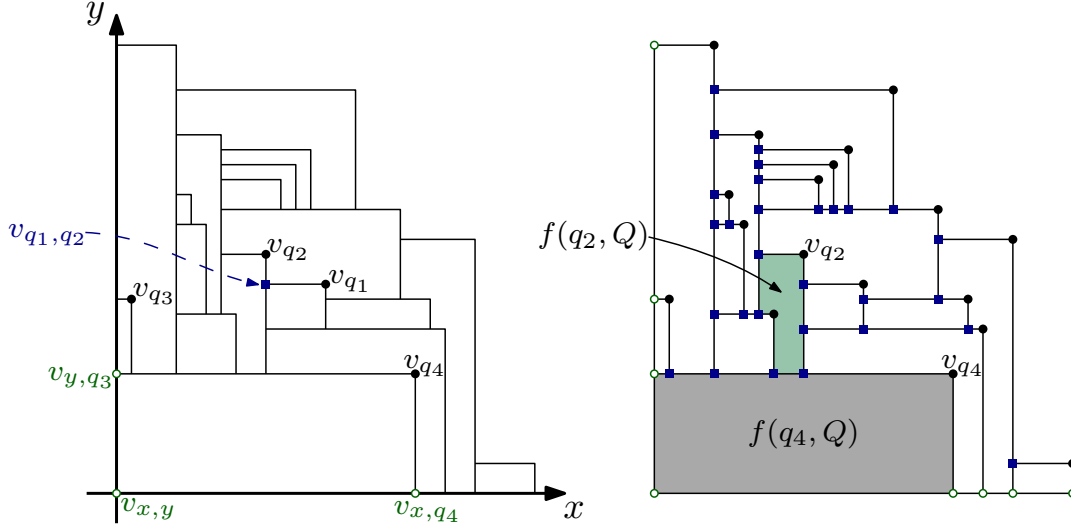


Figure 9: A sample of the different vertices in $G(Q)$ and the faces of $G(Q)$.

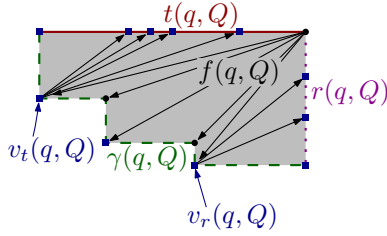


Figure 10: Triangulating a bounded face of $G(Q)$.

several edges of the graph), and a monotone path $\gamma(q, Q)$ from the top, left corner to the bottom, right corner. Such a monotone path $\gamma(q, Q)$ alternates horizontal and vertical segments and has non-decreasing x -coordinates and non-increasing y -coordinates. Let $v_t(q, Q)$ be the first interior vertex of $\gamma(q, Q)$ and let $v_r(q, Q)$ be the last interior vertex of $\gamma(q, Q)$. Note that v_q is the vertex where $t(q, Q)$ and $r(q, Q)$ meet. We add diagonals from v_q to all interior vertices of $\gamma(q, Q)$, diagonals from $v_t(q, Q)$ to all the interior vertices of $t(q, Q)$ and diagonals from $v_r(q, Q)$ to all the interior vertices of $r(q, Q)$. This is the canonical triangulation of the face $f(q, Q)$, and we apply it to each bounded face of $G(Q)$.

The outer face of $G(Q)$ may also have many vertices. We place on top the square σ , with vertices $\{-1, M+1\}^2$. From the vertices at $(-1, -1)$ and $(M+1, M+1)$ we add all possible edges, while keeping planarity. From the vertex $(-1, M+1)$ we add the edges to $(-1, -1)$, to $(M+1, M+1)$, and to the highest vertex on the y -axis. Similarly, from the vertex $(M+1, -1)$ we add the edges to $(-1, -1)$, to $(M+1, M+1)$, and to the rightmost vertex on the x -axis. With such an operation, the outer face is defined by the boundary of the square σ .

Let $T(Q)$ be the resulting geometric, embedded graph, see Figure 11. The graph $T(Q)$ is a triangulation of the square σ with internal vertices. It is easy to see that $G(Q)$ and $T(Q)$ have $O(|Q|)$ vertices and edges. For example, one can argue that $G(Q)$ has $|Q| + 1$ faces and no parallel edges, and the graph $T(Q)$ is a triangulation of $G(Q)$ with 4 additional vertices. To treat some extreme cases, we also define $T(\emptyset) = \sigma$, as a graph, with the diagonal of positive slope.

A polygonal domain is a subset of the plane defined by a polygon where we remove the interior of some polygons, which form holes. The combinatorial complexity of a domain D , denoted by $|D|$, is the number of vertices and edges used to define it. We say that a polygonal curve or a family of polygonal curves in \mathbb{R}^2 is Q -compliant if the edges of the curves are also

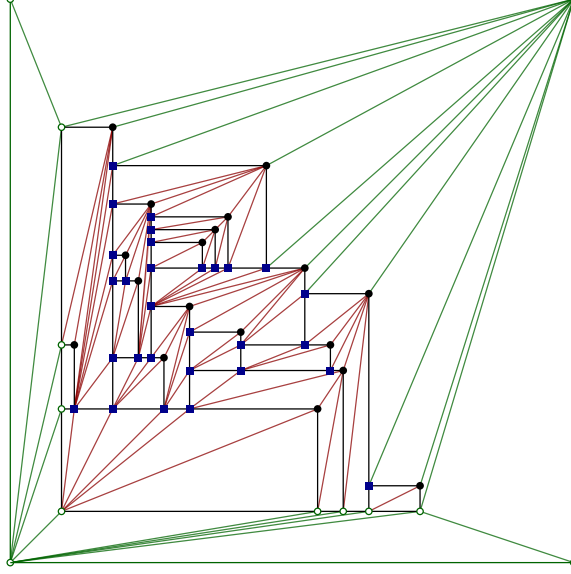


Figure 11: The graph $T(Q)$.

edges of $T(Q)$. A polygonal domain D is Q -**compliant** if its boundary is Q -compliant. Note that a Q -compliant polygonal domain has combinatorial complexity $O(|Q|)$ because the graph $T(Q)$ has $O(|Q|)$ edges.

Consider a set $Q \subseteq P$ and a Q -compliant polygonal curve γ . Let P_γ be the points of P that participate in the definition of the vertices on γ . Thus, if v_q is in γ , we add q to P_γ ; if $v_{q,q'}$ is in γ , we add q and q' to P_γ ; if $v_{x,q}$ is in γ , we add q to P_γ , and so on. Since each vertex on γ contributes $O(1)$ vertices to P_γ , we have $|P_\gamma| = O(|\gamma|)$. For a family Γ of polygonal curves we define $P_\Gamma = \cup_{\gamma \in \Gamma} P_\gamma$. For a polygonal domain D with boundary ∂D we then have $|P_{\partial D}| = O(|D|)$.

Lemma 4.1. *If γ is a Q -compliant polygonal curve then, for each Q' with $P_\gamma \subseteq Q' \subset Q$, the curve γ is also Q' -compliant.*

Proof. For each edge e of $T(Q)$, the edge e is also contained in $T(\tilde{Q})$ for all \tilde{Q} that contain P_e . It follows that $T(Q')$ has all the edges e contained in γ , and thus $T(Q')$ contains γ . \square

We are going to use dynamic programming based on planar separators of $T(Q^*)$ for an optimal solution Q^* . A **valid tuple** to define a subproblem is a tuple (S, D, ℓ) , where $S \subset P$, D is an S -compliant polygonal domain, and ℓ is a positive integer. The tuple (S, D, ℓ) models a subproblem where the points of S are already selected to be part of the feasible solution, D is a S -compliant domain so that we only care about the volume inside the cylinder $D \times \mathbb{R}$, and we can still select ℓ points from $P \cap (D \times \mathbb{R})$. We have two different values associated to each valid tuple, depending on which subsets Q of vertices from $P \cap D$ can be selected:

$$\begin{aligned} \Phi_{\text{free}}(S, D, \ell) &= \max\{\text{VOL}(\mathcal{U}(S \cup Q) \cap (D \times \mathbb{R})) \mid Q \subset P \cap (D \times \mathbb{R}), |Q| \leq \ell\}. \\ \Phi_{\text{comp}}(S, D, \ell) &= \max\{\text{VOL}(\mathcal{U}(S \cup Q) \cap (D \times \mathbb{R})) \mid Q \subset P \cap (D \times \mathbb{R}), |Q| \leq \ell, \\ &\quad D \text{ is } (S \cup Q)\text{-compliant}\}. \end{aligned}$$

Obviously, we have for all valid tuples (S, D, ℓ)

$$\Phi_{\text{comp}}(S, D, \ell) \leq \Phi_{\text{free}}(S, D, \ell).$$

On the other hand, we are interested in the valid tuple (\emptyset, σ, k) , for which we have $\Phi_{\text{free}}(\emptyset, \sigma, k) = \Phi_{\text{comp}}(\emptyset, \sigma, k)$.

We would like to get a recursive formula for $\Phi_{\text{free}}(S, D, \ell)$ or $\Phi_{\text{comp}}(S, D, \ell)$ using planar separators. More precisely, we would like to use a separator in $T(S \cup Q^*)$ for an optimal solution, and then branch on all possible such separators. However, none of the two definitions seem good enough for this. If we would use $\Phi_{\text{free}}(S, D, \ell)$, then we divide into domains that may have too much freedom and the interaction between subproblems gets complex. If we would use $\Phi_{\text{comp}}(S, D, \ell)$, then merging the problems becomes an issue. Thus, we take a mixed route where we argue that, for the valid tuples that are relevant for finding the optimal solution, we actually have $\Phi_{\text{free}} = \Phi_{\text{comp}}$.

We start showing how to compute $\Phi_{\text{comp}}(S, D, \ell)$ in the obvious way. This will be used to solve the base cases of the recursion.

Lemma 4.2. *We can compute $\Phi_{\text{comp}}(S, D, \ell)$ in $O(n^{\ell+2})$ time.*

Proof. We enumerate all the subsets Q of $P \cap D$ with ℓ points, and for each such Q we proceed as follows. We first build $T(S \cup Q)$ and check whether D is contained in the edge set of $T(S \cup Q)$. If it is not, then D is not $(S \cup Q)$ -compliant and we move to the next subset Q . Otherwise, we compute $\mathcal{U}(S \cup Q)$, its restriction to $D \times \mathbb{R}$, and its volume. Standard approaches can be used to do this in $O((|S| + |Q| + |D|)^2) = O(n^2)$ time, for example working with the projection onto the xy -plane. (The actual degree of the polynomial is not relevant.) This procedure enumerates $O(|P|^\ell) = O(n^\ell)$ subsets of P and for each one spends $O(n^2)$ time. The result follows. \square

A **valid partition** π of (S, D, ℓ) is a collection of valid tuples $\pi = \{(S_1, D_1, \ell_1), \dots, (S_t, D_t, \ell_t)\}$ such that

- $S_1 = \dots = S_t = S \cup S_0$ for some set $S_0 \subset P \cap D$;
- $|S_0| = O(\sqrt{|S| + \ell})$;
- the domains D_1, \dots, D_t have pairwise disjoint interiors and $D = \bigcup_i D_i$;
- $\ell = |S_0| + \sum_i \ell_i$; and
- $\ell_i \leq 2\ell/3$ for each $i = 1, \dots, t$.

Let $\Pi(S, D, \ell)$ be the family of valid partitions for the tuple (S, D, ℓ) . We remark that different valid partitions may have different cardinality.

Lemma 4.3. *For each valid tuple (S, D, ℓ) we have*

$$\Phi_{\text{free}}(S, D, \ell) \geq \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \Phi_{\text{free}}(S', D', \ell').$$

Proof. For any valid partition $\pi \in \Pi(S, D, \ell)$, let S_π be the smallest set such that $S' = S \cup S_\pi$ for all tuples $(S', D', \ell') \in \pi$. This means that $S_\pi = S' \setminus S$ for an arbitrary $(S', D', \ell') \in \pi$. For each such tuple $(S', D', \ell') \in \pi$, let $Q^*(S', D', \ell')$ be an optimal solution to $\Phi_{\text{free}}(S', D', \ell')$, and define

$$Q_\pi = S_\pi \cup \bigcup_{(S', D', \ell') \in \pi} Q^*(S', D', \ell').$$

Then from the properties of valid partitions we have

$$|Q_\pi| = |S_\pi| + \sum_{(S', D', \ell') \in \pi} |Q^*(S', D', \ell')| = |S_\pi| + \sum_{(S', D', \ell') \in \pi} \ell' = \ell.$$

Obviously, Q_π is contained in $P \cap D$ because D contains $P \cap D_i$.

We have seen that for each valid partition $\pi \in \Pi(S, D, \ell)$ the set Q_π is a feasible solution considered in the problem $\Phi_{\text{free}}(S, D, \ell)$. Therefore

$$\Phi_{\text{free}}(S, D, \ell) \geq \max_{\pi \in \Pi(S, D, \ell)} \text{VOL}(\mathcal{U}(S \cup Q_\pi) \cap (D \times \mathbb{R})).$$

Using that the interiors of $\{D' \mid (S', D', \ell') \in \pi\}$ are pairwise disjoint, and then using that $S' \cup Q^*(S', D', \ell')$ is contained in $S \cup Q_\pi$ for all $(S', D', \ell') \in \pi$, we obtain

$$\begin{aligned} \Phi_{\text{free}}(S, D, \ell) &\geq \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \text{VOL}(\mathcal{U}(S \cup Q_\pi) \cap (D' \times \mathbb{R})) \\ &\geq \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \text{VOL}(\mathcal{U}(S' \cup Q^*(S', D', \ell')) \cap (D' \times \mathbb{R})). \end{aligned}$$

Since $Q^*(S', D', \ell')$ is optimal for $\Phi_{\text{free}}(S', D', \ell')$, we obtain the desired

$$\Phi_{\text{free}}(S, D, \ell) \geq \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \Phi_{\text{free}}(S', D', \ell'). \quad \square$$

Lemma 4.4. *For each valid tuple (S, D, ℓ) we have*

$$\Phi_{\text{comp}}(S, D, \ell) \leq \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \Phi_{\text{comp}}(S', D', \ell').$$

Proof. Let Q^* be the optimal solution defining $\Phi_{\text{comp}}(S, D, \ell)$. Thus, $Q^* \subseteq P \cap D$ has at most ℓ points, D is $(S \cup Q^*)$ -compliant, and

$$\Phi_{\text{comp}}(S, D, \ell) = \text{VOL}(\mathcal{U}(S \cup Q^*) \cap (D \times \mathbb{R})).$$

Consider the triangulation $T(S \cup Q^*)$. This is a 3-connected planar graph. Recall that the boundary of D is contained in $T(S \cup Q^*)$ because D is $(S \cup Q^*)$ -compliant. Note that $T(S \cup Q^*)$ has $O(|S \cup Q^*|) = O(|S| + \ell)$ vertices.

Assign weight $1/|Q^*|$ to the vertices v_q , $q \in Q^*$, and weight 0 to the rest of vertices in $T(S \cup Q^*)$. The sum of the weights is obviously 1. Because of the cycle-separator theorem of Miller [25], there is a cycle γ in $T(S \cup Q^*)$ with $O(\sqrt{|S| + \ell})$ vertices, such that the interior of γ has at most $2|Q^*|/3$ vertices of Q^* and the exterior of γ has at most $2|Q^*|/3$ vertices of Q^* .

Since γ has $O(\sqrt{|S| + \ell})$ vertices, the set P_γ also has $O(\sqrt{|S| + \ell})$ vertices. Note that $P_\gamma \subseteq S \cup Q^*$. Take $S_0 = P_\gamma \setminus S$, so that $S \cup P_\gamma$ is the disjoint union of S and S_0 . Because of Lemma 4.1, the domain D and the cycle γ are $(S \cup S_0)$ -compliant.

The cycle γ breaks the domain D into at least 2 domains. Let $\mathcal{D} = \{D_1, \dots, D_t\}$ be those domains. Since the boundary of each domain $D_i \in \mathcal{D}$ is contained in $\partial D \cup \gamma$, each domain $D_i \in \mathcal{D}$ is $(S \cup S_0)$ -compliant. For each domain $D_i \in \mathcal{D}$, let $Q_i^* = \{q \in Q^* \setminus (S \cup S_0) \mid v_q \in D_i\}$ and let $\ell_i = |Q_i^*|$. Since the interior of D_i is either in the interior or the exterior of γ , we have $\ell_i \leq 2\ell/3$ for each $D_i \in \mathcal{D}$. Moreover, $|\ell| = |S_0| + \sum_i \ell_i$ because the points q of Q^* that could be counted twice have the corresponding vertex v_q on γ , but then they also belong to $P_\gamma \subset S \cup S_0$ and thus cannot belong to Q_i^* .

The properties we have derived imply that $\pi_\gamma = \{(S \cup S_0, D_i, \ell_i) \mid i = 1, \dots, t\}$ is a valid partition of (S, D, ℓ) , and thus $\pi_\gamma \in \Pi(S, D, \ell)$. Moreover Q_i^* is a feasible solution for the problem $\Phi_{\text{comp}}(S \cup S_0, D_i, \ell_i)$. Indeed, since D_i is $(S \cup S_0)$ -compliant and $(S \cup Q^*)$ -compliant, Lemma 4.1 implies that D_i is also $(S \cup S_0 \cup Q_i^*)$ -compliant.

Note that, for each $(S \cup S_0, D_i, \ell_i)$ in the partition π_γ we have

$$\text{VOL}(\mathcal{U}(S \cup Q^*) \cap (D_i \times \mathbb{R})) = \text{VOL}(\mathcal{U}(S \cup S_0 \cup Q_i^*) \cap (D_i \times \mathbb{R})). \quad (1)$$

Indeed, for a point $q \in Q^* \setminus (S \cup S_0 \cup Q_i^*)$, $\text{BOX}(q)$ may contribute to the union $\mathcal{U}(S \cup P_\gamma \cup Q^*)$, but when projected onto the xy -plane it lies outside the domain D_i because the face $f(q, S \cup Q^*)$ lies outside D_i .

Therefore we obtain

$$\Phi_{\text{comp}}(S, D, \ell) = \text{VOL}(\mathcal{U}(S \cup Q^*) \cap (D \times \mathbb{R})) \leq \sum_i \text{VOL}(\mathcal{U}(S \cup Q^*) \cap (D_i \times \mathbb{R})),$$

where we used $D = \bigcup_i D_i$. With equation (1), and then using that Q_i^* is feasible for $\Phi_{\text{comp}}(S \cup S_0, D_i, \ell_i)$, we get

$$\begin{aligned} \Phi_{\text{comp}}(S, D, \ell) &\leq \sum_i \text{VOL}(\mathcal{U}(S \cup S_0 \cup Q_i^*) \cap (D_i \times \mathbb{R})) \\ &\leq \sum_i \Phi_{\text{comp}}(S \cup S_0, D_i, \ell_i) = \sum_{(S', D', \ell') \in \pi_\gamma} \Phi_{\text{comp}}(S', D', \ell'). \end{aligned}$$

The statement now follows since $\pi_\gamma \in \Pi(S, D, \ell)$. \square

Our dynamic programming algorithm closely follows the inequality of Lemma 4.4. Specifically, we define for each valid tuple (S, D, ℓ) the value

$$\Psi_{\text{comp}}(S, D, \ell) = \begin{cases} \Phi_{\text{comp}}(S, D, \ell) & \text{if } \ell \leq O(\sqrt{k}); \\ \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \Psi_{\text{comp}}(S', D', \ell'), & \text{otherwise.} \end{cases}$$

Lemma 4.5. *For each valid tuple (S, D, ℓ) we have*

$$\Phi_{\text{comp}}(S, D, \ell) \leq \Psi_{\text{comp}}(S, D, \ell) \leq \Phi_{\text{free}}(S, D, \ell).$$

Proof. We show this by induction on ℓ . When $\ell \leq O(\sqrt{k})$, then from the definitions we have

$$\Psi_{\text{comp}}(S, D, \ell) = \Phi_{\text{comp}}(S, D, \ell) \leq \Phi_{\text{free}}(S, D, \ell).$$

This covers the base cases. For larger values of ℓ , we use Lemma 4.4, the induction hypothesis, and the definition of $\Psi_{\text{comp}}(\cdot)$ to derive

$$\begin{aligned} \Phi_{\text{comp}}(S, D, \ell) &\leq \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \Phi_{\text{comp}}(S', D', \ell') \\ &\leq \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \Psi_{\text{comp}}(S', D', \ell') \\ &= \Psi_{\text{comp}}(S, D, \ell). \end{aligned}$$

Also for larger values of ℓ , we use the definition of $\Psi_{\text{comp}}(\cdot)$, the induction hypothesis, and Lemma 4.3, to derive

$$\begin{aligned} \Psi_{\text{comp}}(S, D, \ell) &= \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \Psi_{\text{comp}}(S', D', \ell') \\ &\leq \max_{\pi \in \Pi(S, D, \ell)} \sum_{(S', D', \ell') \in \pi} \Phi_{\text{free}}(S', D', \ell') \\ &\leq \Phi_{\text{free}}(S, D, \ell). \end{aligned} \quad \square$$

Since we know that $\Phi_{\text{free}}(\emptyset, \sigma, k) = \Phi_{\text{comp}}(\emptyset, \sigma, k)$, Lemma 4.5 implies that $\Psi_{\text{comp}}(\emptyset, \sigma, k) = \Phi_{\text{free}}(\emptyset, \sigma, k)$. Hence, it suffices to compute $\Psi_{\text{comp}}(\emptyset, \sigma, k)$ using its recursive definition. In the remainder, we bound the running time of this algorithm.

Theorem 4.6. *In 3 dimensions, VOLUME SELECTION can be solved in time $n^{O(\sqrt{k})}$.*

Proof. We compute $\Psi_{\text{comp}}(\emptyset, \sigma, k)$ using its recursive definition. We need a bound on the number of different subproblems, defined by valid tuples (S, D, ℓ) that appear in the recursion. We will see that there are $n^{O(\sqrt{k})}$ different subproblems.

Starting with $(S_1, D_1, \ell_1) = (\emptyset, \sigma, k)$, consider a sequence of valid tuples $(S_1, D_1, \ell_1), (S_2, D_2, \ell_2), \dots$ such that, for $i \geq 2$, the tuple (S_i, D_i, ℓ_i) appears in some valid partition of $(S_{i-1}, D_{i-1}, \ell_{i-1})$. Because of the properties of valid partitions, we have $\ell_i \leq 2\ell_{i-1}/3$ and $|S_{i-1}| \leq |S_i| \leq |S_{i-1}| + O(\sqrt{|S_i| + \ell_{i-1}})$.

Let i_0 be the first index i with $|S_i| > \ell_i$. Consider first the indices $i < i_0$, where $|S_i| \leq \ell_i$. Then $|S_i| \leq |S_{i-1}| + O(\sqrt{\ell_{i-1}})$ and it follows by induction that

$$\begin{aligned} |S_i| &\leq |S_1| + O(\sqrt{\ell_1}) + O(\sqrt{\ell_2}) + \dots + O(\sqrt{\ell_{i-1}}) \\ &\leq 0 + O\left(\sum_{j<i} \sqrt{\ell_j}\right) \leq O\left(\sum_{j<i} \sqrt{\left(\frac{2}{3}\right)^j \ell_1}\right) \leq O(\sqrt{\ell_1}) \leq O(\sqrt{k}), \end{aligned}$$

where we have used that $\ell_1 = k$. By definition of i_0 , for $i > i_0$ we have $|S_i| \leq |S_{i_0}| + \ell_{i_0} \leq 2|S_{i_0}| = O(\sqrt{k})$. Therefore, for *all* indices i we have $|S_i| = O(\sqrt{k})$.

For each valid tuple that appears in the recursive computation of $\Psi_{\text{comp}}(\emptyset, \sigma, k)$, there is some sequence of valid tuples, as considered before, that contains it. It follows that, for *all* valid tuples (S, D, ℓ) considered through the algorithm we have $|S| = O(\sqrt{k})$.

Let us give an upper bound on the valid tuples (S, D, ℓ) that appear in the computation. There are $n^{O(\sqrt{k})}$ choices for the set S . Once we have fixed S , the domain D has to be S -compliant, and this means that we have to select edges in the triangulated graph $T(S)$. Since $T(S)$ has $O(|S|) = O(\sqrt{k})$ vertices and edges, there are at most $2^{|E(T(S))|} = 2^{O(\sqrt{k})}$ possible choices for D . Finally, we have k options for the value ℓ . Therefore, there are at most

$$n^{O(\sqrt{k})} \cdot 2^{O(\sqrt{k})} \cdot k = n^{O(\sqrt{k})}$$

valid tuples (S, D, ℓ) that appear in the recursion.

We next bound how much time we spend for each tuple. Consider a valid tuple (S, D, ℓ) that appears through the recursion. If $\ell = O(\sqrt{k})$, we compute $\Psi_{\text{comp}}(S, D, \ell)$ using Lemma 4.2 in $n^{O(\ell)} = n^{O(\sqrt{k})}$ time. Otherwise, to compute $\Psi_{\text{comp}}(S, D, \ell)$ we have to iterate over all the valid partitions $\Pi(S, D, \ell)$. There are $n^{O(\sqrt{k})}$ such valid partitions. Indeed, we have to select the subset $S_0 \subset D \cap P$ with $O(\sqrt{k})$ vertices and then the partitioning of D into regions D_1, \dots, D_t that are $(S \cup S_0)$ -compliant. This can be bounded by $n^{O(\sqrt{k})}$. (Alternatively, we can iterate over the $n^{O(\sqrt{k})}$ possible options to define the separating cycle γ used in the proof of Lemma 4.4.)

We conclude that in the computation of $\Psi_{\text{comp}}(\emptyset, \sigma, k)$ we have to consider $n^{O(\sqrt{k})}$ valid tuples and for each one of them computing $\Psi_{\text{comp}}(\cdot)$ takes $n^{O(\sqrt{k})}$ time. The result follows. \square

We only described an algorithm that computes $\text{VOLSEL}(P, k)$, i.e., the maximal volume realized by any size- k subset of P . It is easy to augment the algorithm with appropriate bookkeeping to also compute an actual optimal subset.

5 Efficient Polynomial-Time Approximation Scheme

In this section we design an approximation algorithm for VOLUME SELECTION.

Theorem 5.1. *Given a point set P of size n in $\mathbb{R}_{>0}^d$, $0 \leq k \leq n$, and $0 < \varepsilon \leq 1/2$, we can compute a $(1 \pm \varepsilon)$ -approximation of $\text{VOLSEL}(P, k)$ in time $O(n \cdot \varepsilon^{-d} (\log n + k + 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d}))$. We can also compute a set $S \subseteq P$ of size at most k such that $\mu(S)$ is a $(1 - \varepsilon)$ -approximation of $\text{VOLSEL}(P, k)$ in the same time.*

We also discuss an improvement to time $O(2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d} \cdot n \log n)$ in Section 5.4.

The approach is based on the shifting technique of Hochbaum and Maass [21]. However, there are some non-standard aspects in our application. It is impossible to break the problem into independent subproblems because all the anchored boxes intersect around the origin. We instead break the input into subproblems that are *almost* independent. To achieve this, we use an exponential grid, instead of the usual regular grid with equal-size cells. Alternatively, this could be interpreted as using a regular grid in a log-log plot of the input points.

Throughout this section we need two numbers $\lambda, \tau \approx d/\varepsilon$. Specifically, we define τ as the smallest integer larger than d/ε , and λ as the smallest power of $(1 - \varepsilon)^{-1/d}$ larger than d/ε . We consider a partitioning of the positive quadrant $\mathbb{R}_{>0}^d$ into **regions** of the form

$$R(\bar{x}) := \prod_{i=1}^d [\lambda^{x_i}, \lambda^{x_i+1}) \quad \text{for } \bar{x} = (x_1, \dots, x_d) \in \mathbb{Z}^d.$$

On top of this partitioning we consider a grid, where each grid cell contains $(\tau - 1)^d$ regions and the grid boundaries are thick, i.e., two grid cells do not touch but have a region in between. More precisely, for any offset $\bar{\ell} = (\ell_1, \dots, \ell_d) \in \mathbb{Z}^d$, we define the grid **cells**

$$C_{\bar{\ell}}(\bar{y}) := \prod_{i=1}^d [\lambda^{\tau y_i + \ell_i + 1}, \lambda^{\tau(y_i+1) + \ell_i}) \quad \text{for } \bar{y} = (y_1, \dots, y_d) \in \mathbb{Z}^d.$$

Note that each grid cell indeed consists of $(\tau - 1)^d$ regions, and the space not contained in any grid cell (i.e., the grid boundaries) consists of all regions $R(\bar{x})$ with $x_i \equiv \ell_i \pmod{\tau}$ for *some* $1 \leq i \leq d$.

Our approximation algorithm now works as follows (cf. the pseudocode given below).

(1) Iterate over all grid offsets $\bar{\ell} \in [\tau]^d$. This is the key step of the shifting technique of Hochbaum and Maass [21].

(2) For any choice of the offset $\bar{\ell}$, remove all points not contained in any grid cell, i.e., remove points contained in the thick grid boundaries. This yields a set $P' \subseteq P$ of remaining points.

(3) The grid cells now induce a partitioning of P' into sets P'_1, \dots, P'_m , where each P'_i is the intersection of P' with a grid cell C_i (with $C_i = C_{\bar{\ell}}(\bar{y}^{(i)})$ for some $\bar{y}^{(i)} \in \mathbb{Z}^d$). Note that these grid cell subproblems P'_1, \dots, P'_m are not independent, since any two boxes have a common intersection near the origin, no matter how different their coordinates are. However, we will see that we may treat P'_1, \dots, P'_m as independent subproblems since we only want an approximation.

(4) We discretize by rounding down all coordinates of all points in P'_1, \dots, P'_m to powers of $(1 - \varepsilon)^{1/d}$. We can remove duplicate points that are rounded to the same coordinates. This yields sets $\tilde{P}_1, \dots, \tilde{P}_m$. Note that within each grid cell in any dimension the largest and smallest coordinate differ by a factor of at most $\lambda^{\tau-1}$. Hence, there are at most $\log_{(1-\varepsilon)^{-1/d}}(\lambda^{\tau-1}) = O(\varepsilon^{-2} \log 1/\varepsilon)$ different rounded coordinates in each dimension, and thus the total number of points in each \tilde{P}_i is $O(\varepsilon^{-2} \log 1/\varepsilon)^d$.

(5) Since there are only few points in each \tilde{P}_i , we can precompute all VOLUME SELECTION solutions on each set \tilde{P}_i , i.e., for any $1 \leq i \leq m$ and any $0 \leq k' \leq |\tilde{P}_i|$ we precompute $\text{VOLSEL}(\tilde{P}_i, k')$. We do so by exhaustively enumerating all $2^{|\tilde{P}_i|}$ subsets S of \tilde{P}_i , and for each one computing $\mu(S)$ by inclusion-exclusion in time $O(2^{|S|})$ (see, e.g., [34, 35]). This runs in total time $O(m \cdot 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d}) = O(n \cdot 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d})$.

(6) It remains to split the k points that we want to choose over the subproblems $\tilde{P}_1, \dots, \tilde{P}_m$. As we treat these subproblems independently, we compute

$$V(\bar{\ell}) := \max_{k_1 + \dots + k_m \leq k} \sum_{i=1}^m \text{VOLSEL}(\tilde{P}_i, k_i).$$

³Here we use that λ is a power of $(1 - \varepsilon)^{-1/d}$, to ensure that rounded points are contained in the same cells as their originals.

Note that if the subproblems would be independent, then this expression would yield the exact result. We argue below that the subproblems are sufficiently close to being independent that this expression yields a $(1 - \varepsilon)$ -approximation of $\text{VOLSEL}(\bigcup_{i=1}^m \tilde{P}_i, k)$. Observe that the expression $V(\bar{\ell})$ can be computed efficiently by dynamic programming, where we compute for each i and k' the following value:

$$T[i, k'] = \max_{k_1 + \dots + k_i \leq k'} \sum_{i'=1}^i \text{VOLSEL}(\tilde{P}_{i'}, k_{i'}).$$

The following rule computes this table (see the pseudocode below for further details):

$$T[i, k'] = \max_{0 \leq \kappa \leq \min\{k', |\tilde{P}_i|\}} (\text{VOLSEL}(\tilde{P}_i, \kappa) + T[i-1, k' - \kappa]).$$

(7) Finally, we optimize over the offset $\bar{\ell}$ by returning the maximal $V(\bar{\ell})$.

This finishes the description of the approximation algorithm. In pseudocode, this yields the following procedure.

- (1) Iterate over all offsets $\bar{\ell} = (\ell_1, \dots, \ell_d) \in [\tau]^d$:
 - (2) $P' := P$. Delete any p from P' that is not contained in any grid cell $C_{\bar{\ell}}(\bar{y})$.
 - (3) Partition P' into P'_1, \dots, P'_m , where $P'_i = P' \cap C_i$ for some grid cell C_i .
 - (4) Round down all coordinates to powers of $(1 - \varepsilon)^{1/d}$ and remove duplicates, obtaining $\tilde{P}_1, \dots, \tilde{P}_m$.
 - (5) Compute $H[i, k'] := \text{VOLSEL}(\tilde{P}_i, k')$ for all $1 \leq i \leq m$, $0 \leq k' \leq |\tilde{P}_i|$.
 - (6) Compute $V(\bar{\ell}) := \max_{k_1 + \dots + k_m \leq k} \sum_{i=1}^m \text{VOLSEL}(\tilde{P}_i, k_i)$ by dynamic programming:
 - Initialize $T[i, k'] = 0$ for all $0 \leq i \leq m$, $0 \leq k' \leq k$.
 - For $i = 1, \dots, m$, for $\kappa = 0, \dots, |\tilde{P}_i|$, and for $k' = \kappa, \kappa + 1, \dots, k$:
 - Set $T[i, k'] := \max\{T[i, k'], H[i, \kappa] + T[i-1, k' - \kappa]\}$
 - Set $V(\bar{\ell}) := T[m, k]$.
- (7) Return $\max_{\bar{\ell}} V(\bar{\ell})$.

5.1 Running Time

Step (1) yields a factor $\tau^d = O(\frac{1}{\varepsilon})^d$ in the running time. Since we can compute for each point in constant time the grid cell it is contained in, step (2) runs in time $O(n)$. For the partitioning in step (3), we use a dictionary data structure storing all $\bar{y} \in \mathbb{Z}^d$ with nonempty $P' \cap C_{\bar{\ell}}(\bar{y})$. Then we can assign any point $p \in P'$ to the other points in its cell by one lookup in the dictionary, in time $O(\log n)$. Thus, step (3) can be performed in time $O(n \log n)$. Step (4) immediately works in the same running time. For step (5) we already argued above that it can be performed in time $O(n 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d})$. Finally, from the pseudocode for step (6) we read off a running time of $O(\sum_{i=1}^m |\tilde{P}_i| \cdot k) = O(nk)$. The total running time is thus

$$O\left(n \cdot \varepsilon^{-d} (\log n + k + 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d})\right).$$

5.2 Correctness

The following lemmas show that the above algorithm indeed computes a $(1 \pm O(\varepsilon))$ -approximation of $\text{VOLSEL}(P)$. Reducing ε by an appropriate constant factor then yields a $(1 \pm \varepsilon)$ -approximation.

Lemma 5.2 (Removing grid boundaries). *Let P be a point set and let $0 \leq k \leq |P|$. Remove all points contained in grid boundaries with offset $\bar{\ell}$ to obtain the point set $P_{\bar{\ell}} := P \cap \bigcup_{\bar{y} \in \mathbb{Z}^d} C_{\bar{\ell}}(\bar{y})$. Then for all $\bar{\ell} \in \mathbb{Z}^d$ we have*

$$\text{VOLSEL}(P_{\bar{\ell}}, k) \leq \text{VOLSEL}(P, k),$$

and for some $\bar{\ell} \in \mathbb{Z}^d$ we have

$$\text{VOLSEL}(P_{\bar{\ell}}, k) \geq (1 - \varepsilon) \text{VOLSEL}(P, k).$$

Proof. Since we only remove points, the first inequality is immediate. For the second inequality we use a probabilistic argument. Consider an optimal solution, i.e., a set $S \subseteq P$ of size at most k with $\mu(S) = \text{VOLSEL}(P, k)$. Let $S_{\bar{\ell}} := S \cap P_{\bar{\ell}}$. For a uniformly random offset $\bar{\ell} \in [\tau]^d$, consider the probability that a fixed point $p \in S$ survives, i.e., we have $p \in S_{\bar{\ell}}$. Consider the region $R(\bar{x}) = \prod_{i=1}^d [\lambda^{x_i}, \lambda^{x_i+1})$ containing point p , where $\bar{x} = (x_1, \dots, x_d) \in \mathbb{Z}^d$. Recall that the grid boundaries consist of all regions $R(\bar{x})$ with $x_i \equiv \ell_i \pmod{\tau}$ for some $1 \leq i \leq d$. For a uniformly random $\bar{\ell}$, for fixed i the equation $x_i \equiv \ell_i \pmod{\tau}$ holds with probability $1/\tau$. By a union bound, the probability that at least one of these equations holds for $1 \leq i \leq d$ is at most $d/\tau \leq \varepsilon$ (by definition of τ as the smallest integer larger than d/ε). Hence, p survives with probability at least $1 - \varepsilon$.

Now for each point $q \in \mathcal{U}(S)$ identify a point $s(q) \in S$ dominating q . Since $s(q)$ survives in $S_{\bar{\ell}}$ with probability at least $1 - \varepsilon$, the point q is dominated by $S_{\bar{\ell}}$ with probability at least $1 - \varepsilon$. By integrating over all $q \in \mathcal{U}(S)$ we thus obtain an expected volume of

$$\mathbb{E}_{\bar{\ell}}[\mu(S_{\bar{\ell}})] = \int_{\mathcal{U}(S)} \Pr[q \text{ is dominated by } S_{\bar{\ell}}] dq \geq \int_{\mathcal{U}(S)} (1 - \varepsilon) dq = (1 - \varepsilon) \mu(S).$$

It follows that for some $\bar{\ell}$ we have $\mu(S_{\bar{\ell}}) \geq \mathbb{E}[\mu(S_{\bar{\ell}})] \geq (1 - \varepsilon) \mu(S)$. For this $\bar{\ell}$ we have

$$\text{VOLSEL}(P_{\bar{\ell}}, k) \geq \mu(S_{\bar{\ell}}) \geq (1 - \varepsilon) \mu(S) = (1 - \varepsilon) \text{VOLSEL}(P, k),$$

where the first inequality uses $|S_{\bar{\ell}}| \leq k$ and the definition of VOLSEL as maximizing over all subsets, and the last inequality holds since we picked S as an optimal solution, realizing $\text{VOLSEL}(P, k)$. \square

Lemma 5.3 (Rounding down coordinates). *Let P be a point set, and let \tilde{P} be the same point set after rounding down all coordinates to powers of $(1 - \varepsilon)^{-1/d}$. Then for any k*

$$(1 - \varepsilon) \text{VOLSEL}(P, k) \leq \text{VOLSEL}(\tilde{P}, k) \leq \text{VOLSEL}(P, k).$$

Proof. Let \hat{P} be the set P with all coordinates scaled down by a factor $\alpha := (1 - \varepsilon)^{1/d}$. By a simple scaling invariance, we have $\text{VOLSEL}(\hat{P}, k) = \alpha^d \cdot \text{VOLSEL}(P, k) = (1 - \varepsilon) \text{VOLSEL}(P, k)$. Note that for any point $\tilde{p} \in \tilde{P}$ the corresponding point $p \in P$ dominates \tilde{p} , and the corresponding point $\hat{p} \in \hat{P}$ is dominated by \tilde{p} . Now pick any subset \tilde{S} of \tilde{P} of size k , and let S, \hat{S} be the corresponding subsets of P, \hat{P} . Then we have $\mathcal{U}(\hat{S}) \subseteq \mathcal{U}(\tilde{S}) \subseteq \mathcal{U}(S)$, which implies $\mu(\hat{S}) \leq \mu(\tilde{S}) \leq \mu(S)$, and thus

$$(1 - \varepsilon) \text{VOLSEL}(P, k) = \text{VOLSEL}(\hat{P}, k) \leq \text{VOLSEL}(\tilde{P}, k) \leq \text{VOLSEL}(P, k). \quad \square$$

In the proof of the next lemma it becomes important that we have used the thick grid boundaries, with a separating region, when defining the grid cells.

Lemma 5.4 (Treating subproblems as independent I). *For any offset $\bar{\ell}$, let S_1, \dots, S_m be point sets contained in different grid cells with respect to offset $\bar{\ell}$. Then we have*

$$(1 - \varepsilon) \sum_{i=1}^m \mu(S_i) \leq \mu\left(\bigcup_{i=1}^m S_i\right) \leq \sum_{i=1}^m \mu(S_i).$$

Proof. The second inequality is essentially the union bound. Specifically, for any sets X_1, \dots, X_m the volume of $\bigcup_{i=1}^m X_i$ is at most the sum over all volumes of X_i for $1 \leq i \leq m$. In particular, this statement holds with $X_i = \mathcal{U}(S_i)$, which yields the second inequality.

For the first inequality, observe that we obtain the total volume of all points dominated by $S_1 \cup \dots \cup S_m$ by summing up the volume of all points dominated by S_i but not by any S_j , $j < i$, for each $1 \leq i \leq m$, i.e., we have

$$\mu\left(\bigcup_{i=1}^m S_i\right) = \sum_{i=1}^m \left(\mu(S_i) - \text{VOL}\left(\mathcal{U}(S_i) \cap \bigcup_{j<i} \mathcal{U}(S_j)\right) \right). \quad (2)$$

Now let $C_{\bar{\ell}}(\bar{y}^{(i)})$ be the grid cell containing P_i for $1 \leq i \leq m$, where $\bar{y}^{(i)} = (y_1^{(i)}, \dots, y_d^{(i)}) \in \mathbb{Z}^d$. We may assume that these cells are ordered in non-decreasing order of $y_1^{(i)} + \dots + y_d^{(i)}$. Observe that in this ordering, for any $j < i$ we have $y_t^{(j)} < y_t^{(i)}$ for *some* $1 \leq t \leq d$. Recall that $C_{\bar{\ell}}(\bar{y}) = \prod_{t=1}^d [\lambda^{\tau \cdot y_t + \ell_t + 1}, \lambda^{\tau(y_t + 1) + \ell_t}]$. It follows that each point in $\bigcup_{j<i} \mathcal{U}(S_j)$ has t -th coordinate at most $\delta_t := \lambda^{\tau \cdot y_t + \ell_t}$ for *some* $1 \leq t \leq d$. Setting $D_t := \{(z_1, \dots, z_d) \in \mathbb{R}_{\geq 0}^d \mid z_t \leq \delta_t\}$, we thus have $\bigcup_{j<i} \mathcal{U}(S_j) \subseteq \bigcup_{t=1}^d D_t$, which yields

$$\text{VOL}\left(\mathcal{U}(S_i) \cap \bigcup_{j<i} \mathcal{U}(S_j)\right) \leq \text{VOL}\left(\mathcal{U}(S_i) \cap \bigcup_{t=1}^d D_t\right) \leq \sum_{t=1}^d \text{VOL}(\mathcal{U}(S_i) \cap D_t). \quad (3)$$

Let A be the $(d-1)$ -dimensional volume of the intersection of $\mathcal{U}(S_i)$ with the plane $x_t = 0$. Since all points in S_i have t -th coordinate at least $\lambda^{\tau \cdot y_t + \ell_t + 1} = \lambda \cdot \delta_t$, we have $\mu(S_i) \geq A \cdot \lambda \cdot \delta_t$. Moreover, $\mathcal{U}(S_i) \cap D_t$ has d -dimensional volume $A \cdot \delta_t$. Together, this yields $\text{VOL}(\mathcal{U}(S_i) \cap D_t) \leq \mu(S_i)/\lambda$. With (2) and (3), we thus obtain

$$\mu\left(\bigcup_{i=1}^m S_i\right) \geq \sum_{i=1}^m \left(\mu(S_i) - d \cdot \mu(S_i)/\lambda \right) \geq (1 - \varepsilon) \sum_{i=1}^m \mu(S_i),$$

since $\lambda \geq d/\varepsilon$. □

Lemma 5.5 (Treating subproblems as independent II). *For any offset $\bar{\ell}$, let P_1, \dots, P_m be point sets contained in different grid cells, and $k \geq 0$. Set $P := \bigcup_{i=1}^m P_i$. Then we have*

$$(1 - \varepsilon) \cdot \max_{k_1 + \dots + k_m \leq k} \sum_{i=1}^m \text{VOLSEL}(P_i, k_i) \leq \text{VOLSEL}(P, k) \leq \max_{k_1 + \dots + k_m \leq k} \sum_{i=1}^m \text{VOLSEL}(P_i, k_i).$$

Proof. Consider an optimal solution S of $\text{VOLSEL}(P, k)$ and let $S_i := S \cap P_i$ for $1 \leq i \leq m$. Then by choice of S as an optimal solution, and by Lemma 5.4, we have

$$\text{VOLSEL}(P, k) = \mu(S) = \mu\left(\bigcup_{i=1}^m S_i\right) \leq \sum_{i=1}^m \mu(S_i).$$

Since VOLSEL maximizes over all subsets and $\sum_{i=1}^m |S_i| = |S| \leq k$, we further obtain

$$\sum_{i=1}^m \mu(S_i) \leq \sum_{i=1}^m \text{VOLSEL}(P_i, |S_i|) \leq \max_{k_1 + \dots + k_m \leq k} \sum_{i=1}^m \text{VOLSEL}(P_i, k_i).$$

This shows the second inequality.

For the first inequality, we pick sets S_1, \dots, S_m , where $S_i \subseteq P_i$ for all i and $\sum_{i=1}^m |S_i| \leq k$, realizing $\max_{k_1 + \dots + k_m \leq k} \sum_{i=1}^m \text{VOLSEL}(P_i, k_i) = \sum_{i=1}^m \mu(S_i)$. We then argue analogously:

$$(1 - \varepsilon) \max_{k_1 + \dots + k_m \leq k} \sum_{i=1}^m \text{VOLSEL}(P_i, k_i) = (1 - \varepsilon) \sum_{i=1}^m \mu(S_i) \leq \mu\left(\bigcup_{i=1}^m S_i\right) \leq \text{VOLSEL}(P, k). \quad \square$$

Note that the above lemmas indeed prove that the algorithm returns a $(1 \pm O(\varepsilon))$ -approximation to the value $\text{VOLSEL}(P, k)$. In step (2) we delete the points containing the the grid boundaries, which yields an approximation for some choice of the offset $\bar{\ell}$ by Lemma 5.2. As we iterate over all possible choices for $\bar{\ell}$ and maximize over the resulting volume, we obtain an approximation. In step (4) we round down coordinates, which yields an approximation by Lemma 5.3. Finally, in step (6) we solve the problem $\max_{k_1 + \dots + k_m \leq k} \sum_{i=1}^m \text{VOLSEL}(\tilde{P}_i, k_i)$, which yields an approximation to $\text{VOLSEL}(\bigcup_{i=1}^m \tilde{P}_i, k)$ by Lemma 5.5. All other steps do not change the point set or the considered problem. The final approximation factor is $1 \pm O(\varepsilon)$.

5.3 Computing an Output Set

The above algorithm only gives an approximation for the value $\text{VOLSEL}(P, k)$, but does not yield a subset $S \subseteq P$ of size k realizing this value. However, by tracing the dynamic programming table we can reconstruct the values $k_1 + \dots + k_m \leq k$ with $V(\bar{\ell}) = \sum_{i=1}^m \text{VOLSEL}(\tilde{P}_i, k_i)$. By storing in step (5) not only the values $H[i, k']$ but also corresponding subsets $\tilde{S}_{i, k'} \subset \tilde{P}_i$, we can thus construct a subset $\tilde{S} = \tilde{S}_{1, k_1} \cup \dots \cup \tilde{S}_{m, k_m}$ with $V(\bar{\ell}) = \sum_{i=1}^m \mu(\tilde{S}_{i, k_i})$. Lemma 5.4 now implies that

$$\mu(\tilde{S}) \geq (1 - \varepsilon)V(\bar{\ell}).$$

By storing in step (4) for each rounded point an original point, we can construct a set S corresponding to the rounded points \tilde{S} such that

$$\mu(S) \geq \mu(\tilde{S}) \geq (1 - \varepsilon)V(\bar{\ell}) \geq (1 - O(\varepsilon))\text{VOLSEL}(P, k),$$

and thus S is a subset of P of size at most k yielding a $(1 - O(\varepsilon))$ -approximation of the optimal volume $\text{VOLSEL}(P, k)$.

Note that we do not compute the exact volume $\mu(S)$ of the output set S . Instead, the value $V(\bar{\ell})$ only is a $(1 + O(\varepsilon))$ -approximation of $\mu(S)$. To explain this effect, recall that exactly computing $\mu(T)$ for any given set T takes time $n^{\Theta(d)}$ (under the Exponential Time Hypothesis). As our running time is $O(n^2)$ for any constant d, ε , we cannot expect to compute $\mu(S)$ exactly.

5.4 Improved Algorithm

The following improvement was suggested to us by Timothy Chan. For constant d and ε the algorithm shown above runs in time $O(n(k + \log n))$. The bottleneck for the $O(nk)$ -term is step (6): Given $H_i(k') := \text{VOLSEL}(\tilde{P}_i, k')$ for all $1 \leq i \leq m$, $0 \leq k' \leq |\tilde{P}_i|$, we want to compute

$$\max_{k_1 + \dots + k_m \leq k} \sum_{i=1}^m H_i(k_i).$$

Note that it suffices to compute an $(1 + \varepsilon)$ -approximation to this value, to end up with an $(1 + O(\varepsilon))$ -approximation overall.

This problem is an instance of the *multiple-choice 0/1 knapsack problem*, where we are given a budget W and items $j \in S$ with corresponding weights w_j and profits p_j , as well as a partitioning $S = S_1 \cup \dots \cup S_m$, and the task is to compute the maximum $\sum_{j \in T} p_j$ over all sets $T \subseteq S$ satisfying $\sum_{j \in T} w_j \leq W$ and $|T \cap S_i| = 1$ for all $1 \leq i \leq m$. In order to cast the above problem as an instance of multiple-choice 0/1 knapsack, we simply set $S_i := \{0, 1, \dots, \min\{k, |\tilde{P}_i|\}\}$ and define $p_j := H_i(j)$ and $w_j = j$ for all $j \in S_i$. We also set $W := k$. Note that now the constraint $\sum_{j \in T} w_j \leq W$ corresponds to $k_1 + \dots + k_m \leq k$ and the objective $\sum_{j \in T} p_j$ corresponds to $\sum_{i=1}^m H_i(k_i)$.

For the multiple-choice 0/1 knapsack problem there are known PTAS techniques. In particular, in his Master's thesis, Rhee [29, Section 4.2] claims a time bound of $O(m\varepsilon^{-2} \log(m/\varepsilon) \max_j |S_j| + |S| \log |S|)$. In our case, we have $m \leq n$ and $|S_j| = \min\{k, |\tilde{P}_i|\} + 1 = O(\varepsilon^{-2} \log 1/\varepsilon)^d$. Moreover, $|S| \leq m \cdot \max_j |S_j|$. This yields a time of $O(n \log(n/\varepsilon) \cdot (\varepsilon^{-2} \log 1/\varepsilon)^d)$.

Plugging this solution for step (6) into the algorithm from the previous sections, we obtain time

$$O\left(n \cdot \varepsilon^{-d} (\log n + \log(n/\varepsilon)) \cdot (\varepsilon^{-2} \log 1/\varepsilon)^d + 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d}\right).$$

This can be simplified to $O(n(\log(n/\varepsilon)) \cdot \varepsilon^{-3d} \cdot \log^d(1/\varepsilon) + 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d})$, which is bounded by $O(2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d} \cdot n \log n)$.

6 Conclusions

We considered the volume selection problem, where we are given n points in $\mathbb{R}_{>0}^d$ and want to select k of them that maximize the volume of the union of the spanned anchored boxes. We show: (1) Volume selection is NP-hard in dimension $d = 3$ (previously this was only known when d is part of the input). (2) In 3 dimensions, we design an $n^{O(\sqrt{k})}$ algorithm (the previously best was $\Omega(\binom{n}{k})$). (3) We design an efficient polynomial time approximation scheme for any constant dimension d (previously only a $(1 - 1/e)$ -approximation was known).

We leave open to improve our NP-hardness result to a matching lower bound under the Exponential Time Hypothesis, e.g., to show that in $d = 3$ any algorithm takes time $n^{\Omega(\sqrt{k})}$ and in any constant dimension $d \geq 4$ any algorithm takes time $n^{\Omega(k)}$. Alternatively, there could be a faster algorithm, e.g., in time $n^{O(k^{1-1/d})}$. Finally, we leave open to figure out the optimal dependence on n, k, d, ε of a $(1 - \varepsilon)$ -approximation algorithm.

Moving away from the applications, one could also study volume selection on general axis-aligned boxes in \mathbb{R}^d , i.e., not necessarily anchored boxes. This problem GENERAL VOLUME SELECTION is an optimization variant of Klee’s measure problem and thus might be theoretically motivated. However, GENERAL VOLUME SELECTION is probably much harder than the restriction to anchored boxes, by analogies to the problem of computing an independent set of boxes, which is not known to have a PTAS [1]. In particular, GENERAL VOLUME SELECTION is NP-hard already in 2 dimensions, which follows from NP-hardness of computing an independent set in a family of congruent squares in the plane [18, 22].

Acknowledgements This work was initiated during the Fixed-Parameter Computational Geometry Workshop at the Lorentz Center, 2016. We are grateful to the other participants of the workshop and the Lorentz Center for their support. We are especially grateful to Günter Rote for several discussions and related work.

References

- [1] A. Adamaszek and A. Wiese. Approximation schemes for maximum weight independent set of rectangles. In *Proc. of the 54th IEEE Symp. on Found. of Comp. Science (FOCS)*, pages 400–409. IEEE, 2013.
- [2] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Investigating and exploiting the bias of the weighted hypervolume to articulate user preferences. In *Proc. of the 11th Conf. on Genetic and Evolutionary Computation (GECCO)*, pages 563–570. ACM, 2009.
- [3] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Comp. Science*, 425:75–103, 2012.
- [4] J. Bader. *Hypervolume-based search for multiobjective optimization: theory and methods*. PhD thesis, ETH Zurich, Zurich, Switzerland, 1993.

- [5] F. Barahona. On the computational complexity of Ising spin glass models. *J. of Physics A: Mathematical and General*, 15(10):3241, 1982.
- [6] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold. On the complexity of computing the hypervolume indicator. *IEEE Trans. on Evolutionary Computation*, 13(5):1075–1082, 2009.
- [7] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European J. of Operational Research*, 181(3):1653–1669, 2007.
- [8] K. Bringmann. Bringing order to special cases of Klee’s measure problem. In *Proc. of the 38th Int. Symp. on Mathematical Foundations of Comp. Science (MFCS)*, pages 207–218. Springer, 2013.
- [9] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry*, 43(6):601 – 610, 2010.
- [10] K. Bringmann and T. Friedrich. An efficient algorithm for computing hypervolume contributions. *Evolutionary Computation*, 18(3):383–402, 2010.
- [11] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. *Theoretical Comp. Science*, 425:104–116, 2012.
- [12] K. Bringmann, T. Friedrich, and P. Klitzke. Generic postprocessing via subset selection for hypervolume and epsilon-indicator. In *Proc. of the 13th Int. Conf. on Parallel Problem Solving from Nature (PPSN)*, pages 518–527. Springer, 2014.
- [13] K. Bringmann, T. Friedrich, and P. Klitzke. Two-dimensional subset selection for hypervolume and epsilon-indicator. In *Proc. of the 16th Conf. on Genetic and Evolutionary Comput. (GECCO)*, pages 589–596. ACM, 2014.
- [14] T. M. Chan. A (slightly) faster algorithm for Klee’s measure problem. *Computational Geometry*, 43(3):243–250, 2010.
- [15] T. M. Chan. Klee’s measure problem made easy. In *Proc. of the 54th IEEE Symp. on Found. of Comp. Science (FOCS)*, pages 410–419. IEEE, 2013.
- [16] J. Chen, X. Huang, I. A. Kanj, and G. Xia. Linear FPT reductions and computational lower bounds. In *Proc. of the 36th ACM Symp. on Theory of Computing (STOC)*, pages 212–221. ACM, 2004.
- [17] M. Emmerich, A. H. Deutz, and I. Yevseyeva. A Bayesian approach to portfolio selection in multicriteria group decision making. *Procedia Comp. Science*, 64:993–1000, 2015.
- [18] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Information Processing Lett.*, 12(3):133–137, 1981.
- [19] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem in NP complete. *SIAM J. of Applied Math.*, 32:826–834, 1977.
- [20] A. P. Guerreiro, C. M. Fonseca, and L. Paquete. Greedy hypervolume subset selection in low dimensions. *Evolutionary Computation*, 24(3):521–544, 2016.
- [21] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985.
- [22] H. Imai and T. Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *J. of Algorithms*, 4(4):310–323, 1983.

- [23] J. D. Knowles, D. W. Corne, and M. Fleischer. Bounded archiving using the Lebesgue measure. In *Proc. of the 2003 Congress on Evolutionary Computation (CEC)*, volume 4, pages 2490–2497. IEEE, 2003.
- [24] T. Kuhn, C. M. Fonseca, L. Paquete, S. Ruzika, M. M. Duarte, and J. R. Figueira. Hypervolume subset selection in two dimensions: Formulations and algorithms. *Evolutionary Computation*, 2015.
- [25] G. L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *J. Comput. Syst. Sci.*, 32(3):265–279, 1986.
- [26] J. S. B. Mitchell and M. Sharir. New results on shortest paths in three dimensions. In *Proc. of the 20th ACM Symp. on Computational Geometry (SoCG)*, pages 124–133, 2004.
- [27] N. J. A. Sloane, editor. The on-line encyclopedia of integer sequences. Published electronically at <https://oeis.org>. Visited November 19, 2016.
- [28] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.
- [29] D. Rhee. Faster fully polynomial approximation schemes for knapsack problems, 2015. Master’s thesis. <https://dspace.mit.edu/handle/1721.1/98564>.
- [30] G. Rote, K. Buchin, K. Bringmann, S. Cabello, and M. Emmerich. Selecting k points that maximize the convex hull volume (extended abstract). In *JCDCG3 2016; The 19th Japan Conf. on Discrete and Computational Geometry, Graphs, and Games*, pages 58–60, 9 2016. http://www.jcdcgg.u-tokai.ac.jp/JCDCG3_abstracts.pdf.
- [31] J. A. Storer. On minimal-node-cost planar embeddings. *Networks*, 14(2):181–212, 1984.
- [32] R. Tamassia and I. G. Tollis. Planar grid embedding in linear time. *IEEE Trans. on Circuits and Systems*, 36(9):1230–1234, 1989.
- [33] T. Ulrich and L. Thiele. Bounding the effectiveness of hypervolume-based $(\mu+\lambda)$ -archiving algorithms. In *Learning and Intelligent Optimization*, pages 235–249. Springer, 2012.
- [34] L. While, P. Hingston, L. Barone, and S. Huband. A faster algorithm for calculating hypervolume. *IEEE Trans. on Evolutionary Computation*, 10(1):29–38, 2006.
- [35] J. Wu and S. Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *J. of Mechanical Design*, 123(1):18–25, 2001.
- [36] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. on Evolutionary Computation*, 7(2):117–132, 2003.