Maximum Volume Subset Selection for Anchored Boxes

Karl Bringmann[1], Sergio Cabello[*2], and Michael T. M. Emmerich[3]

1 Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
2 Department of Mathematics, IMFM, Ljubljana, Slovenia; and Department of Mathematics, FMF, University of Ljubljana, Ljubljana, Slovenia
3 Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Leiden, The Netherlands

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. The 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 $\Omega\big(\binom{n}{k}\big)$. We show that:

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

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases geometric optimization, subset selection, hypervolume indicator, Klee's measure problem, boxes, NP-hardness, PTAS

Digital Object Identifier 10.4230/LIPIcs.SoCG.2017.22

1 Introduction

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

$$\textsc{VolSel}(P, k) := \max_{S \subseteq P,\, |S| = k} \textsc{vol}\Big(\bigcup_{p \in S} \textsc{box}(p)\Big),$$

as well as a set $S^* \subseteq P$ of size k realizing this value. Here, $\textsc{vol}$ 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)*

[*] Supported by the Slovenian Research Agency, program P1-0297 and project L7-5459.

33rd International Symposium on Computational Geometry (SoCG 2017).
Editors: Boris Aronov and Matthew J. Katz; Article No. 22; pp. 22:1–22:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

[2, 3, 4, 24, 12, 13]. 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* [34]. Note that with this quality indicator, finding the optimal size-k representation is equivalent to our problem $\textsc{VolSel}(P, k)$. In applications, such bounded-size representations are required in archivers for non-dominated sets [23] and for multicriteria optimization algorithms and heuristics [3, 10, 7].[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 dimension. Finally, in 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 time[2] $O(n^{d/2})$, 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 $\textsc{VolSel}(P, |P|)$).

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 $f(k) \cdot n^{o(k)}$ algorithm under the Exponential Time Hypothesis [16], Klee's measure problem has no $f(d) \cdot n^{o(d)}$ 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].

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 $O\big(\binom{n}{k} k^{d/3}\text{polylog}(k)\big)$ algorithm. For

[1] We 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 $\mathbb{R}^d$.

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

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., [31]), the greedy algorithm for submodular function maximization [27] yields a $(1-1/e)$-approximation of $\textsc{VolSel}(P,k)$. 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 $d = 3$ [28].

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 give a $(1-\varepsilon)$-approximation algorithm running in time $O((n/\varepsilon^d)(\log n + k + 2^{O(\varepsilon^{-2}\log 1/\varepsilon)^d}))$, 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 $p = (p_1, \ldots, p_d)$, $q = (q_1, \ldots, q_d) \in \mathbb{R}^d$, we say that p *dominates* q if $p_i \geq q_i$ for all $1 \leq i \leq d$. For $p = (p_1, \ldots, p_d) \in \mathbb{R}^d_{>0}$, we let $\textsc{box}(p) := [0, p_1] \times \ldots \times [0, p_d]$. Note that $\textsc{box}(p)$ is the set of all points $q \in \mathbb{R}^d_{\geq 0}$ that are dominated by p. A *point set* P is a set of points in $\mathbb{R}^d_{>0}$. We denote the union $\bigcup_{p \in P} \textsc{box}(p)$ by $\mathcal{U}(P)$. The usual Euclidean volume is denoted by $\textsc{vol}$. With this notation, we set

$$\mu(P) := \textsc{vol}(\mathcal{U}(P)) = \textsc{vol}\Big(\bigcup_{p\in P} \textsc{box}(p)\Big) = \textsc{vol}\Big(\bigcup_{p\in P} [0,p_1] \times \ldots \times [0,p_d]\Big).$$

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

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

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

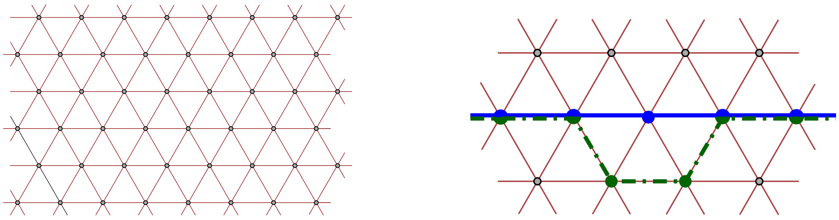

Figure 1 Left: triangular grid Γ. Right: choosing the parity of paths.

3 Hardness in 3 dimensions

We consider the following decision variant of 3-dimensional Volume Selection: Given a triple (P, k, V), where P is a set of points in $\mathbb{R}^3_{>0}$, k is a positive integer and V is a positive real value, 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. 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):

$$\begin{aligned} V(\Gamma) &= \{(i + j \cdot 1/2, j \cdot \sqrt{3}/2) \mid i, j \in \mathbb{N}\}, \\ E(\Gamma) &= \{ab \mid a, b \in V(\Gamma), \text{ the Euclidean distance between } a \text{ and } b \text{ is exactly } 1\}. \end{aligned}$$

We use the problem Independent Set on Induced Triangular Grid: Given a pair (A, ℓ), where A is a subset of $V(\Gamma)$ and ℓ is a positive integer, is there a subset $B \subseteq A$ of ℓ vertices such that no two vertices of B are connected by an edge of $E(\Gamma)$?

▶ **Lemma 3.1.** Independent Set on Induced Triangular Grid *is NP-complete.*

Proof Sketch. Garey and Johnson [19] show that the problem Vertex Cover is NP-complete for planar graphs of degree at most 3, which implies that Independent Set is NP-complete for planar graphs of degree at most 3.

Given a planar graph G of degree at most 3, we construct an orthogonal drawing of G on a square grid of polynomial size [29, 30] and transform it into a drawing of G on Γ. Rescaling and rerouting, we get a graph H that is an induced subgraph of Γ, and a subdivision of G where each edge of G is path in H with an even number of interior vertices. See Figure 1, right, to see how to choose the parity of the path. If $\alpha(G)$ is the size of the largest independent set in G, and each edge uv of G is represented by a path with $2k_{uv}$ internal vertices, then $\alpha(H) = \alpha(G) + \sum_{uv \in E(G)} k_{uv}$. Indeed, we can obtain H from G by repeatedly replacing an edge by a 3-edge path, and any such replacement increases the size of the largest independent set by exactly 1. ◀

3.2 The point set

Let $m \geq 3$ be an arbitrary integer and consider the point set P_m defined by $P_m = \{(x, y, z) \in \mathbb{N}^3 \mid x + y + z = m\}$, see Figure 2. Standard induction shows that the set P_m has $(m-1)(m-2)/2$ points and that $\mu(P_m) = m(m-1)(m-2)/6$.

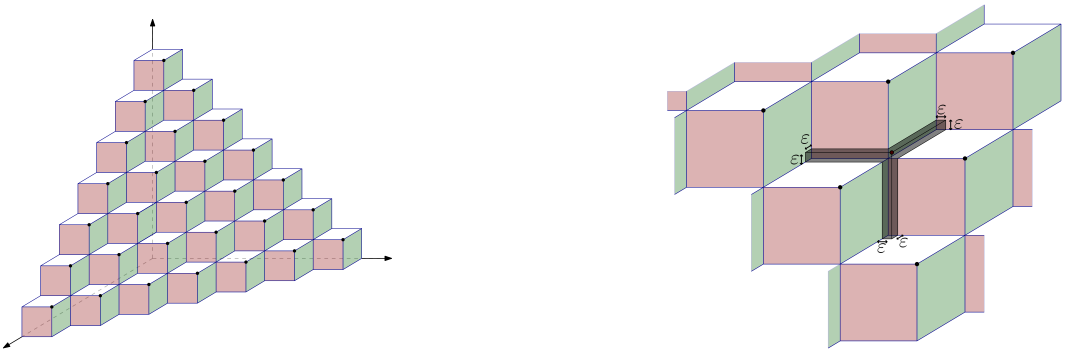

Figure 2 Left: the point set P_m and the boxes $\textsc{box}(p)$, with $p \in P_m$. Right: the point $q = p + \Delta_\varepsilon$ and the set $\textsc{diff}(q)$.

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 2, right. Let us define the set $Q_m = \{p + \Delta_\varepsilon \mid p \in P_{m-1}\}$. 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

$$\textsc{diff}(q) \;=\; \mathcal{U}(P_m \cup \{q\}) \setminus \mathcal{U}(P_m) \;=\; \Bigg(\bigcup_{p \in P_m \cup \{q\}} \textsc{box}(p) \Bigg) \setminus \Bigg(\bigcup_{p \in P_m} \textsc{box}(p) \Bigg).$$

Note that $\textsc{diff}(q)$ 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 2, right. The sets and the parameter ε are selected to have the following properties.

▶ **Lemma 3.2.** *The following holds.*

- *If $Q' \subseteq Q_m$ and the sets $\textsc{diff}(q)$, 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 $Q' \subseteq Q_m$ and Q' contains two points q_0 and q_1 such that $\textsc{diff}(q_0)$ and $\textsc{diff}(q_1)$ intersect, then $\mu(P_m \cup Q') < \mu(P_m) + |Q'| \cdot (3\varepsilon^2 + \varepsilon^3)$.*
- *If P' is a subset of P_m such that $P_m \setminus P'$ is non-empty, then $\mu(P' \cup Q_m) < \mu(P_m)$.*

3.3 The reduction

We can define naturally a graph T_m on the set Q_m by using the intersection of the sets $\textsc{diff}(\cdot)$. 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 $\textsc{diff}(q)$ and $\textsc{diff}(q')$ intersect, see Figure 3. Simple geometry shows that T_m is isomorphic to a part of the triangular grid Γ, up to scaling. 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 $\{\textsc{diff}(q) \mid q \in Q'\}$ are pairwise disjoint.

▶ **Theorem 3.3.** *The problem* Volume Selection *is NP-complete in* 3 *dimensions.*

Proof. 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. 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 $V = \frac{m(m-1)(m-2)}{6} + \ell \cdot (3\varepsilon^2 + \varepsilon^3)$. Then we can show that (A, ℓ) is a yes

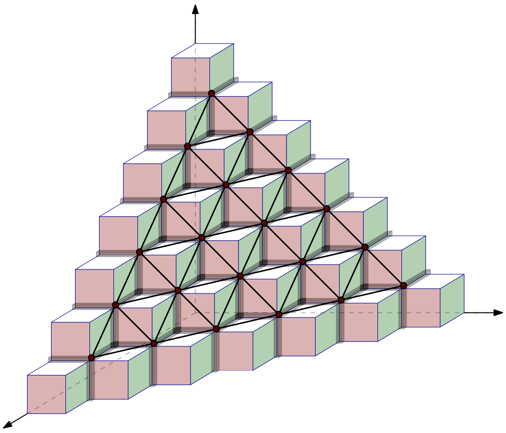

Figure 3 The graph T_m for $m = 9$.

instance for Independent Set on Induced Triangular Grid if and only if (P, k, V) is a yes instance for 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 $\{\textsc{diff}(q) \mid q \in Q_m(B)\}$ are pairwise disjoint. Lemma 3.2 then implies that

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

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 $\mu(P_m \cup Q_m(B)) = V$ and thus (P, k, V) is a yes instance for Volume Selection.

Assume now that (P, k, V) is a yes instance for 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 3.2, 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 $\mu(Q) \geq V$. Thus it must be that $Q_m(B)$ is an independent set in T_m. It follows that $B \subset A$ has size ℓ and is an independent set in Γ, and thus (A, ℓ) is a yes instance for Independent Set on Induced Triangular Grid. ◀

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.

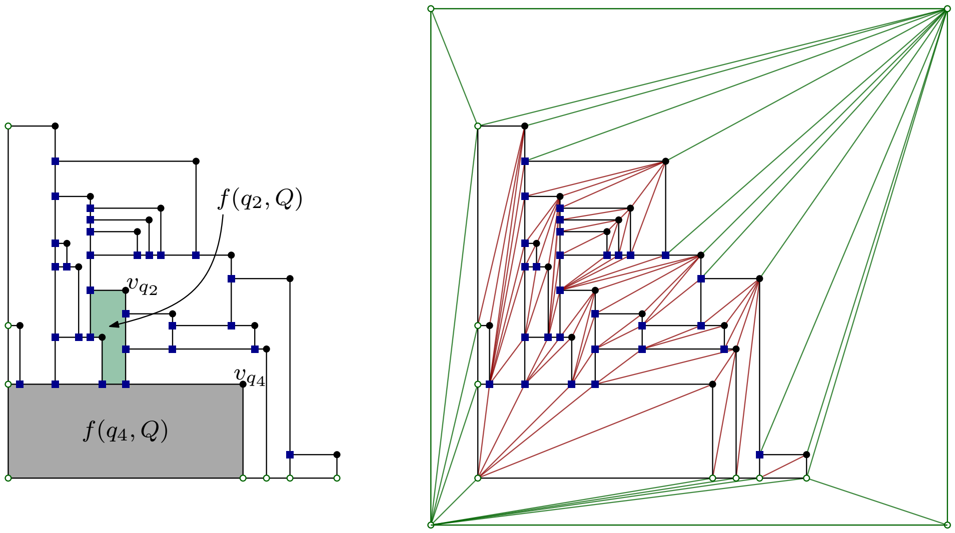

Figure 4 The graphs $G(Q)$ (left) and $T(Q)$ (right).

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. 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 $G(Q)$; see Figure 4, left. We consider $G(Q)$ as a geometric, embedded graph where each vertex is a point and each edge is a horizontal or vertical straight-line segment on the xy-plane. The projection of each point $q \in Q$ defines a vertex, which we denote by v_q. 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 $\{(x, y, z) \in \mathbb{R}^3 \mid z = q_z\}$. 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*, see Figure 4 right. We add all possible edges from the top rightmost vertex v_q, then all possible edges from the bottom leftmost vertex, and finally all edges from the left bottom-most vertex. 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 $\{-1, M+1\}^2$, and triangulate in some systematic way. Let $T(Q)$ be the resulting geometric, embedded graph, see Figure 4, right. 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.

A polygonal domain is a subset of the plane defined by a polygon where we remove the interior of some polygons, which form holes. A polygonal domain D is ***Q-compliant*** if its boundary is contained in the edge set of $T(Q)$. Note that a Q-compliant polygonal domain has $O(|Q|)$ edges because the graph $T(Q)$ has $O(|Q|)$ edges.

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:

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

Obviously, for all valid tuples (S, D, ℓ) we have $\Phi_{\text{comp}}(S, D, \ell) \le \Phi_{\text{free}}(S, D, \ell)$. On the other hand, we are interested in the valid tuple $(\emptyset, \sigma, k)$, 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 $\Phi_{\text{free}} = \Phi_{\text{comp}}$.

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

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

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

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

Proof Sketch. For the first inequality, we show that, for each $\pi \in \Pi(S, D, \ell)$, joining solutions to the subproblems $\Phi_{\text{free}}(\cdot)$ defined by $\{(S', D', \ell') \mid (S', D', \ell') \in \pi\}$ gives a feasible solution for the problem $\Phi_{\text{free}}(S, D, \ell)$.

For the second inequality, we consider an optimal solution $Q^* \subseteq P \cap D$ with at most ℓ points for the problem $\Phi_{\text{comp}}(S, D, \ell)$. The triangulation $T(S \cup Q^*)$ is a 3-connected planar graph and the boundary of D is contained in $T(S \cup Q^*)$ because D is $(S \cup Q^*)$-compliant. We now use the cycle-separator theorem of Miller [25] to split the vertices of Q^*: There is a cycle γ in $T(S \cup Q^*)$ of length $O(\sqrt{|S| + \ell})$ 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^*. Using this cycle separator we can build a valid partition $\pi_\gamma \in \Pi(S, D, \ell)$ such that $Q^* \cap D'$ is a feasible solution to each $(S', D', \ell') \in \pi_\gamma$. For the correctness argument, we use an easy monotonicity property of being Q-compliant, which we skip in this short version. We then have

$$\Phi_{\text{comp}}(S, D, \ell) \;\leq\; \sum_{(S',D',\ell') \in \pi_\gamma} \Phi_{\text{comp}}(S', D', \ell'),$$

and the second inequality follows. ◀

Our dynamic programming algorithm closely follows the inequalities of Lemma 4.1. Specifically, we define for each valid tuple (S, D, ℓ) the value

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

Standard induction on ℓ using Lemma 4.1 implies the following property.

▶ **Lemma 4.2.** *For each valid tuple (S, D, ℓ) we have*

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

Since we know that $\Phi_{\text{free}}(\emptyset, \sigma, k) = \Phi_{\text{comp}}(\emptyset, \sigma, k)$, Lemma 4.2 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 4.3.** *In 3 dimensions,* Volume Selection *can be solved in time $n^{O(\sqrt{k})}$.*

Proof Sketch. We compute $\Psi_{\text{comp}}(\emptyset, \sigma, k)$ using its recursive definition. The base cases, where $\ell = O(\sqrt{k})$, can be solved in $n^{O(\ell)} = n^{O(\sqrt{k})}$ time using simple enumeration of all size-ℓ subsets.

Starting with $(S_1, D_1, \ell_1) = (\emptyset, \sigma, k)$, consider a sequence of valid tuples (S_1, D_1, ℓ_1), (S_2, D_2, ℓ_2), ... such that, for $i \geq 2$, the tuple (S_i, D_i, ℓ_i) appears in some valid partition of $(S_{i-1}, D_{i-1}, \ell_{i-1})$. By the properties of valid partitions, we have $\ell_i \leq 2\ell_{i-1}/3$ and $|S_{i-1}| \leq |S_i| \leq |S_{i-1}| + O(\sqrt{|S_i| + \ell_{i-1}})$. It follows that the sequence $\ell_1, \ell_2, \ldots$ decreases geometrically, from which one can deduce that $|S_i| = O(\sqrt{k})$ for all i. This means that there are $n^{O(\sqrt{k})}$ valid tuples (S, D, ℓ) that appear in the recursive calls. The same bound can be shown for the number of valid partitions in each step. ◀

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 $\mathbb{R}^d_{>0}$, $0 \leq k \leq n$, 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.*

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 $\mathbb{R}^d_{>0}$ into ***regions*** of the form

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

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***

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

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* $1 \le i \le d$.

5.1 Description of the algorithm

Our approximation algorithm works as follows.

(1) Iterate over all grid offsets $\bar{\ell} \in [\tau]^d$. This is the key step of the shifting technique [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. Call the remaining points $P' \subseteq P$.

(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, as shown below treating $P'_1, \dots, P'_m$ as independent subproblems still yields an approximation.

(4) We discretize by rounding down all coordinates of all points in $P'_1, \dots, P'_m$ to powers of[3] $(1-\varepsilon)^{1/d}$. 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 $1 \le i \le m$ and any $0 \le k' \le |\tilde{P}_i|$ we precompute $\textsc{VolSel}(\tilde{P}_i, k')$. 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., [32, 33]). This runs in total time $O(m \cdot 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d}) = O(n \cdot 2^{O(\varepsilon^{-2} \log 1/\varepsilon)^d})$.

[3] Here 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.

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

$$V(\bar{\ell}) := \max_{k_1+\ldots+k_m \le k} \sum_{i=1}^{m} \textsc{VolSel}(\tilde{P}_i, k_i).$$

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 $\textsc{VolSel}(\bigcup_{i=1}^{m} \tilde{P}_i, k)$. 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:

$$T[i,k'] = \max_{k_1+\ldots+k_i \le k'} \sum_{i'=1}^{i} \textsc{VolSel}(\tilde{P}_{i'}, k_{i'}).$$

The following rule computes this table:

$$T[i,k'] = \max_{0 \le \kappa \le \min\{k', |\tilde{P}_i|\}} \big(\textsc{VolSel}(\tilde{P}_i, \kappa) + T[i-1, k'-\kappa]\big).$$

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

In pseudocode, this yields the following procedure:

```
(1) Iterate over all offsets $\bar{\ell} = (\ell_1, \ldots, \ell_d) \in [\tau]^d$:
    (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 $P'_1, \ldots, P'_m$, 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
        $\tilde{P}_1, \ldots, \tilde{P}_m$.
    (5) Compute $H[i,k'] := \textsc{VolSel}(\tilde{P}_i, k')$ for all $1 \le i \le m$, $0 \le k' \le |\tilde{P}_i|$.
    (6) Compute $V(\bar{\ell}) := \max_{k_1+\ldots+k_m \le k} \sum_{i=1}^{m} \textsc{VolSel}(\tilde{P}_i, k_i)$ by dynamic programming.
(7) Return $\max_{\bar{\ell}} V(\bar{\ell})$.
```

5.2 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 $O\big(n2^{O(\varepsilon^{-2}\log 1/\varepsilon)^d}\big)$. Finally, step (6) can be implemented in time $O(\sum_{i=1}^{m} |\tilde{P}_i| \cdot k) = O(nk)$. The total running time is thus $O\big(n \cdot \varepsilon^{-d}\big(\log n + k + 2^{O(\varepsilon^{-2}\log 1/\varepsilon)^d}\big)\big)$.

5.3 Correctness

Combining the following lemmas we show that the above algorithm indeed computes a $(1 \pm O(\varepsilon))$-approximation of $\textsc{VolSel}(P)$.

▶ **Lemma 5.2** (Removing grid boundaries). *Let P be a point set and let $0 \le k \le |P|$. 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 $\textsc{VolSel}(P_{\bar{\ell}}, k) \le \textsc{VolSel}(P, k)$, and for* some *$\bar{\ell} \in \mathbb{Z}^d$ we have $\textsc{VolSel}(P_{\bar{\ell}}, k) \ge (1-\varepsilon)\textsc{VolSel}(P, k)$.*

Proof Sketch. 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 $\mu(S) = \textsc{VolSel}(P, k)$. Let $S_{\bar\ell} := S \cap P_{\bar\ell}$. For a uniformly random offset $\bar\ell \in [\tau]^d$, the probability that a fixed point $p \in S$ does *not* survive, i.e., we have $p \notin S_{\bar\ell}$ is at most $d/\tau \le \varepsilon$. 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

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

It follows that for some $\bar\ell$ we have $\mu(S_{\bar\ell}) \ge \mathbb{E}[\mu(S_{\bar\ell})] \ge (1-\varepsilon)\mu(S)$. For this $\bar\ell$ we have $\textsc{VolSel}(P_{\bar\ell}, k) \ge (1-\varepsilon)\textsc{VolSel}(P, k)$. ◀

▶ **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*

$$(1-\varepsilon)\textsc{VolSel}(P,k) \le \textsc{VolSel}(\tilde P, k) \le \textsc{VolSel}(P,k).$$

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*

$$(1-\varepsilon)\sum_{i=1}^m \mu(S_i) \le \mu\Big(\bigcup_{i=1}^m S_i\Big) \le \sum_{i=1}^m \mu(S_i).$$

Proof Sketch. The second inequality is the union bound applied to $\mathcal{U}(S_1), \dots, \mathcal{U}(S_m)$.

For the first inequality, we can decompose $\bigcup_{i=1}^m \mathcal{U}(S_i)$ to get

$$\mu\Big(\bigcup_{i=1}^m S_i\Big) \;=\; \textsc{vol}\left(\bigcup_{i=1}^m \mathcal{U}(S_i)\right) \;=\; \sum_{i=1}^m \left(\mu(S_i) - \textsc{vol}\Big(\mathcal{U}(S_i) \cap \bigcup_{j<i} \mathcal{U}(S_j)\Big)\right). \tag{1}$$

Now let $C_{\bar\ell}(\bar y^{(i)})$ be the grid cell containing P_i for $1 \le i \le m$, 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* $1 \le t \le d$. 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* $1 \le t \le d$. Setting $D_t := \{(z_1, \dots, z_d) \in \mathbb{R}^d_{\ge 0} \mid z_t \le \delta_t\}$, we thus have $\bigcup_{j<i} \mathcal{U}(S_j) \subseteq \bigcup_{t=1}^d D_t$, which yields

$$\textsc{vol}\Big(\mathcal{U}(S_i) \cap \bigcup_{j<i} \mathcal{U}(S_j)\Big) \le \textsc{vol}\Big(\mathcal{U}(S_i) \cap \bigcup_{t=1}^d D_t\Big) \le \sum_{t=1}^d \textsc{vol}\big(\mathcal{U}(S_i) \cap D_t\big). \tag{2}$$

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 $\mu(S_i) \ge A \cdot \lambda \cdot \delta_t$. Moreover, $\mathcal{U}(S_i) \cap D_t$ has d-dimensional volume $A \cdot \delta_t$. Together, this yields $\textsc{vol}(\mathcal{U}(S_i) \cap D_t) \le \mu(S_i)/\lambda$. With (1) and (2), and using that $\lambda \ge d/\varepsilon$, we thus obtain

$$\mu\Big(\bigcup_{i=1}^m S_i\Big) \ge \sum_{i=1}^m \big(\mu(S_i) - d \cdot \mu(S_i)/\lambda\big) \ge (1-\varepsilon)\sum_{i=1}^m \mu(S_i).$$ ◀

Leveraging the above lemma to VolSel yields the following.

▶ **Lemma 5.5** (Treating subproblems as independent II)**.** *For any offset $\bar{\ell}$, let $P_1, \ldots, P_m$ be point sets contained in different grid cells, and $k \geq 0$. Then we have*

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

Note that the above lemmas indeed prove that the algorithm returns a $(1 \pm O(\varepsilon))$-approximation to the value $\textsc{VolSel}(P, k)$. 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 $\max_{k_1+\ldots+k_m \leq k} \sum_{i=1}^{m} \textsc{VolSel}(\tilde{P}_i, k_i)$, which yields an approximation to $\textsc{VolSel}(\bigcup_{i=1}^{m} \tilde{P}_i, k)$ by Lemma 5.5. All other steps do not change the point set or the considered problem.

5.4 Computing an Output Set

The above algorithm, as described, only gives an approximation for the value $\textsc{VolSel}(P, k)$. However, by tracing the dynamic programming table we can reconstruct a subset S of P of size at most k yielding a $(1 - O(\varepsilon))$-approximation of the optimal volume $\textsc{VolSel}(P, k)$.

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.

6 Conclusions

We considered the volume selection problem, where we are given n points in $\mathbb{R}^d_{>0}$ 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 $d = 2$ (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*, 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 *Int. Symp. on Mathematical Foundations of Comp. Science*, 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 *Int. Conf. on Parallel Problem Solving from Nature*, 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 2014 Conf. on Genetic and Evolutionary Comput.*, 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*, pages 124–133, 2004.

27 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.

28 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`.

29 J. A. Storer. On minimal-node-cost planar embeddings. *Networks*, 14(2):181–212, 1984.

30 R. Tamassia and I. G. Tollis. Planar grid embedding in linear time. *IEEE Trans. on Circuits and Systems*, 36(9):1230–1234, 1989.

31 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.

32 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.

33 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.

34 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.