Tolerated Tverberg Partitions: An Algorithmic Approach

Masterarbeit

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin
Arbeitsgruppe Theoretische Informatik

März 2013

von

Yannik Stein

Betreuer: Prof. Dr. Wolfgang Mulzer

Contents

1 Introduction

As tolerated Tverberg partitions have not been studied in the context of algorithmics so far, this thesis focuses on the development of algorithms that compute such partitions.

Tolerated Tverberg partitions are of interest due to their relationship with a generalization of the median to multi-dimensional data. The median for a set $P \subset \mathbb{R}^d$ is a number that divides the set into two parts of equal size. For this reason, it is often used in divide & conquer algorithms. Especially in the field of statistics, the median is also well known to have benign properties such as scale invariance and tolerance against outliers. Several generalizations for multi-dimensional data exist. One of the most important is the centerpoint theorem by Radon [10]. Given a set $P \subset \mathbb{R}^d$, a point $c \in \mathbb{R}^d$ is called a *centerpoint* if each closed half-space containing c contains at least $|P|/(d+1)$ points of P. Teng proved in his Ph.D. thesis [13] that testing whether a given point is a centerpoint is coNP-complete if the dimensionality is part of the input. However, Tverberg [15] proved the existence of a subset of centerpoints that have polynomial-time checkable witnesses. These witnesses are called Tverberg partitions. A *Tverberg partition* $\mathcal{T}$ of a point set $P \subset \mathbb{R}^d$ is a partition of P into disjoint sets whose convex hulls have a nonempty intersection. The points within that intersection are called *Tverberg points*. Each half-space that contains a point within the intersection has to contain at least one point of each element of the Tverberg partition. Tverberg's theorem states that each point set $P \subset \mathbb{R}^d$ of size at least $(m-1)(d+1)+1$ admits the computation of a Tverberg partition of size m. This implies the existence of a Tverberg partition of size $|P|/(d+1)$ for each point set $P \subset \mathbb{R}^d$, which proves each Tverberg point of that partition to be a centerpoint.

García Colín [5] generalized Tverberg's theorem to *tolerated* Tverberg partitions. A t-tolerated Tverberg partition for a point set $P \subseteq \mathbb{R}^d$ remains a Tverberg partition even if up to t points are removed from P. She conjectured the existence of a t-tolerated Tverberg partition of size m for all point sets $P \subset \mathbb{R}^d$ of size at least $(t+1)(m-1)(d+1)+1$. For $t = 0$, this coincides with the bound in Tverberg's theorem. This conjecture was restated by Montejano and Oliveros [8] along with proofs of several other tolerated versions of Helly-type theorems and finally proved by Soberón and Strausz [12]. Soberón and Strausz also conjectured this bound to be tight. Besides the naive algorithm, which checks all possible partitions of an input set, no algorithm is currently known that computes a tolerated Tverberg partition.

As said in the beginning, medians or more generally centerpoints, are appreciated for their "robustness" with respect to outliers. While the meaning of robustness is a rather intuitive one, meaning the value changes not "much" by changing "some" parts of the input, tolerated Tverberg partitions define robustness formally with respect to the number of changed points.

In contrast to tolerated Tverberg partitions, several algorithms are known that compute centerpoints and Tverberg partitions. The best known algorithm to compute a centerpoint has expected running time $\mathcal{O}(|P|^{d-1})$ [2]. For the case of Tverberg partitions, the best known exact algorithm runs in $|P|^{\mathcal{O}(d^2)}$ time [1]. Miller & Sheehy [7] published the first approximation algorithm for Tverberg partitions in arbitrary dimensions by derandomizing a Monte-Carlo approximation algorithm for centerpoints by Clarkson et al. [3]. This algorithm computes a Tverberg partition of size $|P|/(d+1)^2$ in $n^{\mathcal{O}(\log d)}$ time. Later, Mulzer and Werner [9] developed an algorithm with linear running time in the size of P, which returns a Tverberg partition of size $|P|/4(d+1)^3$ in $d^{\mathcal{O}(\log d)}n$ time.

Related decision problems of the centerpoint theorem and Tverberg's theorem were studied by Teng [13] and Werner [18]. Although those decision problems could be proved to be hard, these results shed only little light on the complexity of actually computing a centerpoint or a Tverberg partition. The complexity for these problems is still unknown if the dimensionality is part of the input. It is widely believed, that these problems cannot be solved in polynomial time. As tolerated Tverberg partitions are a generalization of Tverberg partitions, all complexity results about the problem to compute Tverberg partitions can also be applied to the problem to compute tolerated Tverberg partitions. However, there a currently no complexity results known outside the special case of untolerated Tverberg partitions.

This thesis starts by presenting fundamental definitions and theorems in the context of Tverberg partitions and revisit both approximation algorithms for the untolerated Tverberg problem, with the aim to adapt the techniques developed by Miller & Sheehy as well as by Mulzer & Werner in the following parts. To get a better understanding of the problem, in chapter 3 we study the special case of computing t-tolerated Tverberg partitions of size 2 in one dimension, then try to extend the results to t-tolerated Tverberg partitions of arbitrary size in one dimension and later to multiple dimensions. We also study the implications of these results on the bound by Soberón and Strausz to find arguments that support or reject the tightness-conjecture.

In the second part of the thesis, we are interested in reusing the approximation algorithms for the untolerated Tverberg problem that were presented in chapter 2. To achieve this, we focus on the development of approximation preserving reductions that lead for every approximation algorithm for the untolerated Tverberg problem to an approximation algorithm for the tolerated Tverberg problem and compare concrete results by applying them to the algorithm by Miller & Sheehy as well as to the algorithm by Mulzer & Werner. Although at the moment there is no algorithm known that computes t'-tolerated Tverberg partitions for some constant $t' > 0$, in this context it is also worth exploring if it is possible to benefit from such algorithms.

In the last part, we take a look at the complexity of a related decision problem that is similar to the decision problems studied by Teng [13].

2 Fundamentals

Before we start to analyze the tolerated Tverberg problem, we revisit basic terms and theorems that will be used throughout the thesis. We also present both known deterministic approximation algorithms for the untolerated Tverberg problem by Miller & Sheehy [7] and Mulzer & Werner [9]. Although we can show that both algorithms cannot be used directly as approximation algorithms for the tolerated Tverberg problem, they provide useful techniques that can be adapted in our context, as shown in the following chapters.

We omit the proofs for the theorems in the first section since most of them are well-known results in discrete geometry. For their proofs and more thorough discussions, we refer the reader to the book by Matoušek [6] or the original publications.

2.1 Basic Definitions and Theorems

The median is a well-known statistical concept for one-dimensional data. It splits the upper from the lower half of a set of numbers. In comparison to the mean, outliers and false data have less impact on the value of the median. However, "less impact" has no well-defined meaning in a mathematical context. We will return to this point at the end of this section when we formally define the concept of tolerance.

Definition 2.1 (Median)**.** Let $P \subset \mathbb{R}$ be a set of real numbers. A number $m \in \mathbb{R}$ is called a *median* of P if

$$|\{p \mid p \in P, p \leq m\}| \geq |P|/2 \wedge |\{p \mid p \in P, p \geq m\}| \geq |P|/2$$

We are interested in an analogue concept for multi-dimensional data. Let $P \subset \mathbb{R}^d$ be a point set. In a first attempt, we might take a look at the point $m' \in \mathbb{R}^d$, where the ith value of m' is a median for P projected onto the ith dimension. An example is shown in Figure 2.1. Each point within the dashed rectangle fulfills our criteria. As we said in the beginning, the median splits the data into two parts of equal size. In our example, the point m' only splits the data equally for the case of axis parallel hyperplanes. The non-axis parallel hyperplane h separates m' completely from P, without containing any point from P in the same half-space as m'. A better suited criteria to assess the belonging of a point to the center of a point set is the *Tukey depth*, as it takes every hyperplane into consideration and not only axis-parallel ones.

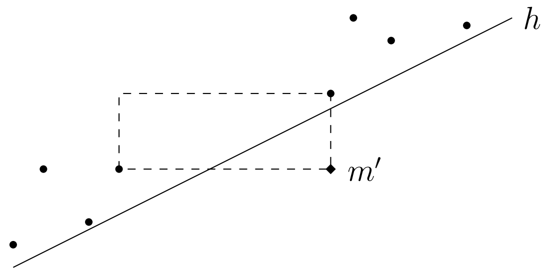

Figure 2.1: Median in Each Dimension

Definition 2.2 (Tukey Depth [6])**.** Let $P \subset \mathbb{R}^d$ be a point set and $c \in \mathbb{R}^d$ a point. We say c has *Tukey depth* r or just c has *depth* r with respect to P if every closed half-space h^- containing c contains also at least r points of P.

Applying this criteria to our example, we obtain a depth of 0 for our candidate point m'. Thus, with regard to the new criteria, m' is not in the center at all.

In one dimension, we know that for every set of real numbers there exists a median. It is of interest to develop a similar result for the Tukey depth. That is, a result that states the existence of a point of some depth k for every point set $P \subset \mathbb{R}^d$, where k is depending only on the size and dimensionality of P. This result is well known as centerpoint theorem.

Theorem 2.3 (Centerpoint Theorem [10])**.** *Given a point set $P \subset \mathbb{R}^d$. Then there exists a point $c \in \mathbb{R}^d$ that has depth at least $\lceil |P|/(d+1) \rceil$ with respect to P. The point c is called a centerpoint of P.*

The centerpoint theorem is known to be sharp, so in general there exist point sets that do not admit the computation of points of depth strictly higher than $\lceil |P|/(d+1) \rceil$.

Teng [13] proved in his Ph.D. thesis that testing whether a given point is a centerpoint is coNP-complete if the dimensionality is part of the input. Especially in the context of Monte-Carlo algorithms, this result is problematic. However, Tverberg [15] proved the existence of centerpoints that have polynomial-time checkable witnesses. These witnesses are called *Tverberg partitions.*

Definition 2.4 (Tverberg Partition [15]). Given a point set $P \subset \mathbb{R}^d$. We call a partition of P into sets $T_1, T_2, \ldots, T_m$ a *Tverberg partition* of size m if $\bigcap_{i=1}^m \mathsf{conv}(T_i) \neq \emptyset$. Every point within the intersection is called a *Tverberg point* of depth m.

The size m of a Tverberg partition is directly connected to the depth of a corresponding Tverberg point c: Each half-space h^- that contains c has to intersect the convex hull of each element of the partition, since c is contained in the intersection of all convex hulls. Therefore, at least one point of each element of the partition is also contained in h^-, so c has depth at least m.

Theorem 2.5 (Tverberg's Theorem [15, 16, 17, 11]). *Given a point set $P \subset \mathbb{R}^d$ of size at least $(m-1)(d+1)+1$. Then there exists a Tverberg partition of P of size m.*

As in the case of the centerpoint theorem, this theorem was shown to be tight. If the size of the Tverberg partition $\mathcal{T}$ is 2, we call $\mathcal{T}$ a *Radon* partition and every Tverberg point of this partition a *Radon* point.

An example is given in Figure 2.2. Each dashed triangle represents an element of the partition. The point in the intersection of all convex hulls, c, is a Tverberg point of this partition. Since the size of this partition is 3 and the number of points is 9, c is a centerpoint for this point set.

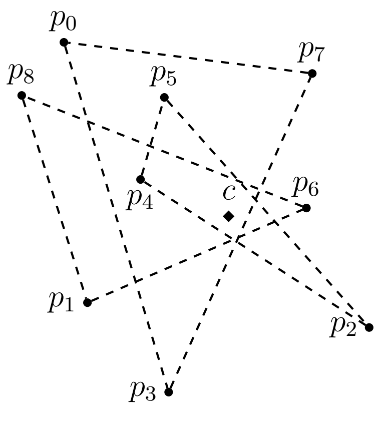

Figure 2.2: Tverberg Partition $\mathcal{T} = \{\{p_2, p_4, p_5\}, \{p_1, p_6, p_8\}, \{p_0, p_7, p_3\}\}$

If we want to prove that a partition $\mathcal{T}$ is a Tverberg partition, we have to show that the intersection of the convex hulls of the elements of $\mathcal{T}$ is nonempty. Helly's Theorem [6] turns out to be very handy in this situation is, as it limits the number of elements we have to look at at once.

Theorem 2.6 (Helly's Theorem [6])**.** *Let $C_1, C_2, \ldots, C_k \subset \mathbb{R}^d$ be convex sets. If $k \geq d+1$ and every $d+1$ of these sets intersect, then the intersection of all convex sets is nonempty.*

In the later chapters, we refer to the problem of computing any Tverberg partition for a given point set as *Tverberg problem.*

Problem 2.7 (Tverberg Problem)**.** Given a point set P of size at least $(m-1)(d+1)+1$. Compute a Tverberg partition of size m for P.

As noted in the beginning of this section, the robustness of the median against outliers and false data is not well defined. We formalize this by introducing the concept of tolerance in the context of Tverberg partitions. A t-tolerated Tverberg partitions remains a Tverberg partition even if up to t input points are removed. In a practical context, we can think of these points as false data or data that is changed after construction of the Tverberg partition.

Definition 2.8 (Tolerated Tverberg Partition [5])**.** Given a point set $P \subset \mathbb{R}^d$. A partition of P into sets $T_1, T_2, \ldots, T_m$ is called a *t-tolerated* Tverberg partition of size m if for every subset $R \subseteq P$ of size at most t, the partition $\{T_1 \setminus R, T_2 \setminus R, \ldots, T_m \setminus R\}$ is a Tverberg partition for $P \setminus R$.

Note that we do not define a t-tolerated Tverberg point, since in general there is no point that remains a Tverberg point after the removal of any t points. We formally prove this in the next section.

Soberón and Strausz [12] generalized Theorem 2.5 by proving the following bound for tolerated Tverberg partitions.

Theorem 2.9 (Soberón-Strausz Bound [12])**.** *Let P be a point set of size at least $(t+1)(m-1)(d+1)+1$. Then there exists a t-tolerated Tverberg partition for P of size m.*

Soberón and Strausz conjectured this bound to be tight. Until now, it is an open question whether this is true. We therefore define the *tolerated* Tverberg problem slightly different from Problem 2.7 by not constraining the size of the input point set but enabling the algorithms to return $\perp$ if it is not possible to compute any t-tolerated Tverberg partition of the desired size.

For each developed algorithm, we will give concrete bounds on the size of the input set that are needed for the algorithms to work properly. If the input size is strictly lower than this bound, we implicitly assume the algorithms to return $\perp$. For the case of the developed approximation algorithms, it is possible that these algorithms return $\perp$ even if there exists a tolerated Tverberg partition of the input set.

Problem 2.10 (Tolerated Tverberg Problem)**.** Given a point set P and a tolerance parameter $t \in \mathbb{R}^d$. Compute a t-tolerated Tverberg partition for P of maximum size if one exists, or output $\perp$.

2.2 Observations

Lemma 2.11. *Let $P \subset \mathbb{R}^d$ be a set. A point $c \in \mathbb{R}^d$ has depth $t+1$ with respect to P if and only if for all subsets $R \subset P$ of size at most t, c is contained in $\mathsf{conv}(P \setminus R)$.*

Proof. "$\Rightarrow$": We prove this direction by contraposition. Suppose R is a subset of P of size at most t such that $c \notin \mathsf{conv}(P \setminus R)$. Let h be a hyperplane that separates c from $\mathsf{conv}(P \setminus R)$ and let h^+ denote the closed half-space that contains c and is bounded by h. Since h separates c from $\mathsf{conv}(P \setminus R)$, the intersection $(P \setminus R) \cap h^+$ has to be empty. Thus, h^+ contains c and at most $|R| = t$ points from P. This implies that c has depth at most $t < t+1$ with respect to P.

"$\Leftarrow$": Let h^+ be some half-space that contains c and let R denote the set $h^+ \cap P$. Then c is not contained in $\mathsf{conv}(P \setminus R)$. If the size of R is less or equal to t, this would be a contradiction to c being contained in $\mathsf{conv}(P \setminus R')$ for all subsets R' of P of size at most t. This implies that c has depth at least $t+1$, as claimed. ■

Proposition 2.12. *Let $d \in \mathbb{N}$ be the dimension. Then there exists a t-tolerated Tverberg partition $\mathcal{T}$ for some set $P \subset \mathbb{R}^d$ such that*

$$\bigcap_{R \subset P : |R| \leq t} \ \bigcap_{T_i \in \mathcal{T}} \mathsf{conv}(T_i \setminus R) = \emptyset$$

Proof. Suppose for the sake of contradiction that for each t-tolerated Tverberg partition there exists a point that is contained in the intersection of the convex hulls of the elements even if any t points are removed from P. Let $P \subset \mathbb{R}^d$ be of size $(t+1)(m-1)(d+1)+1$. Assume without loss of generality that $t > 0$. Theorem 2.9 guarantees the existence of a t-tolerated Tverberg partition $\mathcal{T}$ of size m for P. By our assumption, there exists a point c that is contained in $\bigcup_{T_i \in \mathcal{T}} \mathsf{conv}(T_i \setminus R)$ for all subsets R of size at most t. In particular, c is contained in the convex hull of each element T_i ($i \in \{1, 2, \ldots, m\}$) of $\mathcal{T}$ even if any R points are removed. Lemma 2.11 then implies that c has depth $t+1$ with respect to each element of $\mathcal{T}$. Thus, c has depth $m(t+1)$ with respect to P. However, the centerpoint theorem guarantees only the existence of a point that has depth

$$\left\lceil \frac{(t+1)(m-1)(d+1)+1}{d+1} \right\rceil = (t+1)(m-1)+1 \overset{t>0}{<} m(t+1)$$

This is a contradiction to the tightness of the centerpoint theorem. ■

2.3 Deterministic Approximation Algorithms for the Untolerated Tverberg Problem

In this section, two different approximation algorithms for the untolerated Tverberg problem are presented. We discuss the main idea of each algorithm and prove selected statements. We also take a close look at the properties of these algorithms in an tolerated setting.

2.3.1 Approximation Algorithm by Miller & Sheehy

Let $P \subset \mathbb{R}^d$ be the input point set. The algorithm follows a divide & conquer strategy. We partition P into small sets for which we can obtain trivial Tverberg partitions and then repeatedly combine them to obtain Tverberg partitions of greater size. The crucial part of this approach is the way in which we obtain a Tverberg partition of greater size for the entire point set out of Tverberg partitions for disjoint subsets. Miller & Sheehy developed the following lemma for this task.

Lemma 2.13 ([3]). *Given $d+2$ Tverberg partitions $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_{d+2}$ of disjoint point sets and of size m. Let q_i be a Tverberg point of $\mathcal{T}_i$ and P_i the point set for which $\mathcal{T}_i$ is a partition. Then the Radon point of $\{q_1, q_2, \dots, q_{d+2}\}$ has depth at least $2m$ with respect to P.*

Proof. Let $T_{i,j}$ be the jth element of $\mathcal{T}_i$. Let (Q_1, Q_2) be a Radon partition of $Q = \{q_1, q_2, \dots, q_{d+2}\}$ and $r \in \mathsf{conv}(Q_1) \cap \mathsf{conv}(Q_2)$ a Radon point of Q. Fix some $j \in \{1, 2, \dots, m\}$. We show that r is contained in $\mathsf{conv}(\bigcup_{i=1}^{d+2} T_{i,j})$. Because $T_{i,j}$ is part of the Tverberg partition for q_i, we know by definition that $q_i \in \mathsf{conv}(T_{i,j})$. Since $r \in \mathsf{conv}(\bigcup_{q_i \in Q_k})$ for $k \in \{1, 2\}$, r is contained in $\mathsf{conv}(\bigcup_{q_i \in Q_k} T_{i,j})$. Putting everything together, we know that $r \in \bigcap_{j=1}^{m} \mathsf{conv}(\bigcup_{q_i \in Q_k} T_{i,j})$ for $k \in \{1, 2\}$. By our assumption, all elements of the Tverberg partitions are disjoint. Since the point r is contained in the convex hull of $2m$ disjoint sets, r has depth $2m$. ■

Before we take a look at the concrete algorithm, let us discuss an important optimization to the constructive proof of Lemma 2.13. Let Q be the set $\{q_1, q_2, \dots, q_{d+1}\}$ and let $\mathcal{T}_i = \{T_{i,1}, T_{i,2}, \dots, T_{i,m}\}$ be the Tverberg partition for which q_i is a Tverberg point. Since the bound in Tverberg's theorem is sharp, in general the number of points in each Tverberg partition $\mathcal{T}_i$ is at least $(m-1)(d+1)+1$. Thus, the number of points in the Tverberg partition that is constructed in the proof of Lemma 2.13 is in general at least $(d+1)((m-1)(d+1)+1)$. However, Tverberg's theorem guarantees the existence of Tverberg partitions of size $2m$ for much smaller sets of size at least $(2m-1)(d+1)+1$. By identifying points in the constructed Tverberg partition by Lemma 2.13 that are not needed for the convex hulls of the elements of that partition to have a nonempty intersection, we can remove them and fed back into the algorithm to create more Tverberg partitions that again can be combined. In this way, we virtually increase the size of the input set and thus improve the size of the returned Tverberg partition. We refer to this step as *pruning*. Carathéodory's theorem helps us to identify superfluous points.

Theorem 2.14 (Carathéodory's theorem [6]). *Let $P \subset \mathbb{R}^d$ be a set and $x \in \mathbb{R}^d$ a point that is contained in the convex hull of P. Then there exists a subset $P' \subseteq P$ of size at most $d+1$ whose convex hull contains x.*

Given a Tverberg partition $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$, we fix a Tverberg point c and then apply Carathéodory's theorem to identify $d+1$ points T'_i in each element T_i of $\mathcal{T}$ whose convex hull contains c. Since the convex hull of each of these subsets T'_i contains c, we know that $\bigcap_{i \in \{1,2,\dots,m\}} \mathsf{conv}(T'_i) \neq \emptyset$. We can thus limit the number of points in the partition constructed by Lemma 2.13 to $2m(d+1)$, a great improvement compared to $(d+1)((m-1)(d+1)+1)$.

We are now ready to discuss the complete algorithm. We will focus on the main idea of each step and refer the reader to the original publication for a thorough analysis. In line 2 of Algorithm 2.1, we handle the base case. If the input point set is of small enough size, we just return a trivial Tverberg partition for P of size one, which is a good enough approximation for input sets of small size. In lines 3–6, we recursively compute $d+1$ Tverberg partition of subsets of P. The removal of the used points for these partitions in line 6 guarantees that the subsets are disjoint. The pruning in line 8 enables us to choose subsets of size $|P|/2$ instead of $|P|/(d+1)$ as the number of points in the partition returned by the recursive call is smaller. The analysis by Miller & Sheehy proves the following bounds on the size of the returned partition and on the total running time.

Algorithm 2.1: Iterated-Tverberg [7]

```
input : Point set $P \subset \mathbb{R}^d$
1 if $|P| \leq 2(d+1)^2$ then
2     return $\{P\}$;
3 foreach $i \in \{1, 2, \ldots, d+1\}$ do
4     $P' \leftarrow$ choose any $\lceil n/2 \rceil$ points from $P$;
5     $\mathcal{T}_i \leftarrow$ Iterated-Tverberg($P'$);
6     $P \leftarrow P \setminus (\bigcup_{T_j^{(i)} \in \mathcal{T}_i} T_j^{(i)})$;
7 $\mathcal{T} \leftarrow$ combine $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{d+1}$ using Lemma 2.13;
8 Prune $\mathcal{T}$ using Carathéodory's theorem;
9 return $\mathcal{T}$;
```

Theorem 2.15 ([3])**.** *Given a point set $P \in \mathbb{R}^d$. Algorithm 2.1 computes a Tverberg partition of size $\lceil |P|/(d+1)^2 \rceil$ for P in $n^{\mathcal{O}(\log d)}$ time.*

Let us now take a look at these results in the context of tolerated Tverberg partitions. We start with the pruning. Let $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ be a Tverberg partition and let $\mathcal{T}' = \{T'_1, T'_2, \ldots, T'_m\}$ be the result of $\mathcal{T}$ after pruning. That is, we haven chosen some point $c \in \bigcup_{i \in \{1,2,\ldots,m\}} \mathsf{conv}(T_i)$ and identified a subset T'_i of at most $d+1$ points whose convex hull contains c for each element T_i of the partition $\mathcal{T}$. Since c is contained in the convex hull of each element of $\mathcal{T}'$, the intersection of the convex hulls of all elements in $\mathcal{T}'$ has to contain at least c. Carathéodory's theorem does not guarantee c to be contained in the convex hulls of the pruned elements T'_i if we remove points. More formally, it could be that $c \notin \mathsf{conv}(T'_i \setminus R)$ if $T'_i \cap R \neq \emptyset$. A simple example for this case is c being an element of the interior of the simplex spanned by T'_i. Thus, if points are removed, the convex hulls of the elements of $\mathcal{T}'$ are not guaranteed to have a nonempty intersection, so $\mathcal{T}'$ is an untolerated Tverberg partition. This is independent of the tolerance of the initial unpruned Tverberg partition $\mathcal{T}$.

However, as the pruning was only an optimization, we could just skip this step. Unfortunately, a similar problem arises by applying Lemma 2.13. An essential requirement in the proof of this lemma is, that each point q_i is a Tverberg point for $\mathcal{T}_i$, otherwise an intersection of the constructed t-tolerated Tverberg partition cannot be guaranteed. However, if q_i is a Tverberg point of $\mathcal{T}_i$, this does not guarantee that q_i is still a Tverberg point after the removal of some point of Q_i. This is clear if $\mathcal{T}_i$ is untolerated, as the removal of one point could even lead to an empty intersection of the convex hulls of the elements

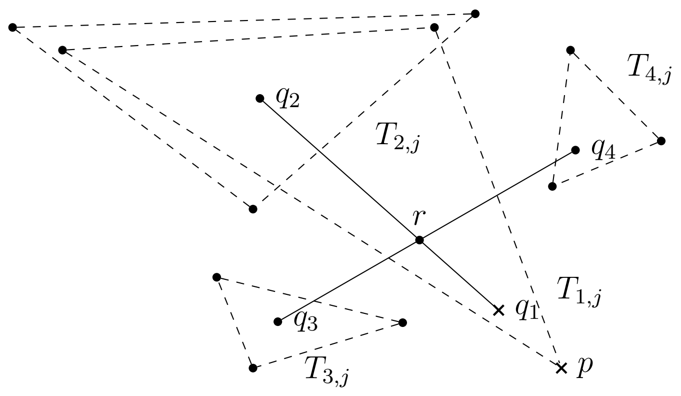

Figure 2.3: Combination of Tverberg partitions using Lemma 2.13 in two dimensions

in $\mathcal{T}_i$. It makes no difference whether $\mathcal{T}_i$ is of some tolerance $t > 0$ or not, as tolerance only guarantees an intersection but states nothing about specific points that are contained in the intersection (cf. Proposition 2.12).

Figure 2.3 shows an example in which the Tverberg partition constructed using Lemma 2.13 is untolerated. The dashed lines indicate the convex hulls of the jth elements of $d + 2 = 4$ Tverberg partitions, where q_i is a Tverberg point of $\mathcal{T}_i$ ($i \in \{1, 2, 3, 4\}$). The Radon partition for $\{q_1, q_2, q_3, q_4\}$ is represented with solid lines. In this example, $T_{1,j} \cup T_{2,j}$ and $T_{3,j} \cup T_{4,j}$ would be elements of the returned partition. If $p \in P_1$ is removed, then $\mathsf{conv}(T_{1,j} \cup T_{2,j} \setminus \{p\}) \cap \mathsf{conv}(T_{3,j} \cup T_{4,j}) = \emptyset$. Thus, the constructed Tverberg partition is untolerated.

2.3.2 Approximation Algorithm by Mulzer & Werner

The algorithm by Miller & Sheehy always operates in the vector space of the input point set P. In contrast to this approach, Mulzer & Werner's algorithm [9] computes a Tverberg partition of the entire input set projected onto a low dimensional subspace and then lifts the computed partition to the original dimensionality. We will only discuss the simplified version of this algorithm. We refer the reader to the original publication for the improved algorithm.

The crucial part of this approach is the way in which we obtain a Tverberg partition for a point set $P \subseteq \mathbb{R}^d$ out of a Tverberg partition of P projected onto some subspace. Mulzer & Werner developed the following lemma to achieve this task.

Lemma 2.16 (Lifting Argument [9]). *Given a point set $P \subseteq \mathbb{R}^d$. Let h be a hyperplane in $\mathbb{R}^d$ and c' a Tverberg point of depth m for the projection of P onto h with a corresponding Tverberg partition $\mathcal{T}'$. Then we can find a Tverberg point $c \in \mathbb{R}^d$ of depth $m/2$ for P and a corresponding Tverberg partition in time $|P|$.*

Proof. Let p_i be the ith point of the input point set and let p'_i denote the projection of p_i onto h. Let further be l the line through c' that is orthogonal to h. By our assumption, $\mathcal{T}' = \{T'_1, \ldots, T'_m\}$ is a Tverberg partition of $P' = \{p'_1, p'_2, \ldots, p'_{|P|}\}$. Let Q_i be the set of points in P that are projected to the points in T'_i, that is $Q_i = \{p_j \mid p'_j \in T'_i\}$. Since c' is a Tverberg point of $\mathcal{T}'$, we know by definition that $c \in \mathsf{conv}(T'_i)$ for all $i \in \{1, 2, \ldots, m\}$. As the points of P were projected orthogonally onto h, l has to have a nonempty intersection with the convex hulls of the sets $Q_1, Q_2, \ldots, Q_m$. Let q_i be some point in $\mathsf{conv}(Q_i) \cap l$ and let $\mathcal{T}_q$ be a Tverberg partition of size $m/2$ for $Q = \{q_1, q_2, \ldots, q_m\}$. A Tverberg partition of this size exists for Q since all points q_i are contained in the one-dimensional affine subspace l and Tverberg's theorem guarantees for all point sets of size m in one dimension the existence of a Tverberg partition of size $m/2$. We then claim $\mathcal{T} = \{\bigcup_{q_i \in T_{q,j}} Q_i \mid T_{q,j} \in \mathcal{T}_q\}$ to be the desired Tverberg partition of size $m/2$ for P.

We first check that $\mathcal{T}$ is indeed a partition of P. By construction, each Q_i contains all points whose projections are contained in the ith element of $\mathcal{T}_q$. Thus the set $\{Q_1, Q_2, \ldots, Q_m\}$ forms a partition of P. Each element of $\mathcal{T}$ is the union of a subset of $\{Q_1, Q_2, \ldots, Q_m\}$ and each Q_i is a subset of exactly one element in $\mathcal{T}$. As a partition of P remains a partition of P if we replace some elements by their union, $\mathcal{T}$ is a partition of P.

Let T_j and T_k be two elements of $\mathcal{T}$. We now prove that the intersection of $\mathsf{conv}(T_j)$ and $\mathsf{conv}(T_k)$ is nonempty. By construction of $\mathcal{T}$, we know that both sets $\{q_i \mid Q_i \subseteq T_j\}$ and $\{q_i \mid Q_i \subseteq T_k\}$ are elements of a Tverberg partition of Q. Thus, $\mathsf{conv}(\{q_i \mid Q_i \subseteq T_j\}) \cap \mathsf{conv}(\{q_i \mid Q_i \subseteq T_k\}) \neq \emptyset$. Since each point q_i is contained in the convex hull of the corresponding set Q_i, the convex hulls $\mathsf{conv}(\{q_i \mid Q_i \subseteq T_j\})$ and $\mathsf{conv}(\{q_i \mid Q_i \subseteq T_k\})$ are contained in $\mathsf{conv}(T_j)$ and $\mathsf{conv}(T_k)$. Therefore, $\mathsf{conv}(T_j)$ and $\mathsf{conv}(T_k)$ have a nonempty intersection. Helly's theorem now implies that $\bigcap_{T_i \in \mathcal{T}} \mathsf{conv}(T_i) \neq \emptyset$, or equivalently that $\mathcal{T}$ is a Tverberg partition. Using the same arguments, we can also prove that each Tverberg point of $\mathcal{T}_q$ is a Tverberg point of the constructed Tverberg partition $\mathcal{T}$.

The only remaining point is the running time in which we can compute $\mathcal{T}$. We can construct the sets $Q_1, Q_2, \ldots, Q_m$ in linear time in the size of P as they form a partition of P. To compute a Tverberg partition of Q, we can use the fact that the points in Q are contained in a one-dimensional affine subspace. Thus, we just need to compute the median of Q and then partition Q into pairs $\{q^-, q^+\}$, where q^- is less than the median and q^+ is greater than the median. The computation of the median can be carried out in $\mathcal{O}(m)$ using the algorithm by Blum et al. [4, Chapter 9]. The partitioning of Q into pairs takes also linear time in the size of Q. This results in a total running time of $\mathcal{O}(|P|)$ as claimed. ■

Algorithm 2.2: Simple-Lifting-Argument [9]

```
input : Point set $P \subset \mathbb{R}^d$
if $d = 1$ then
    $m \leftarrow$ median of $P$;
    $\mathcal{T} \leftarrow$ Partition of $P$ into pairs $\{p^-, p^+\}$, where $p^- \leq m$ and $p^+ \geq m$;
    return $(m, \mathcal{T})$;
else
    $h \leftarrow$ some hyperplane in $\mathbb{R}^d$;
    $P' \leftarrow$ projection of $P$ onto $h$;
    $(c', \mathcal{T}') \leftarrow$ simple-lifting-argument($P'$);
    $(c, \mathcal{T}) \leftarrow$ lift $(c', \mathcal{T}')$ using Lemma 2.16;
    Prune $\mathcal{T}$ using Carathéodory's theorem;
    return $(c, \mathcal{T})$;
```

Algorithm 2.2 is based on this lemma. In lines 1–4, we handle the base case. If the dimensionality is one, we use the median algorithm by Blum et al. to obtain a Tverberg partition for P as described in the proof of Lemma 2.16. If the dimensionality d is greater than one, we project P onto some hyperplane h in $\mathbb{R}^d$ and then obtain recursively a Tverberg partition for the projected points in lines 6–8. This partition is lifted to a Tverberg partition for the input point set in line 9 using Lemma 2.16.

Theorem 2.17 ([9])**.** *Given a point set $P \subset \mathbb{R}^d$. Then Algorithm 2.2 computes a Tverberg partition of size $\lceil n/2^d \rceil$ for P in $d^{\mathcal{O}(1)}|P|$ time.*

The size of the computed Tverberg partition by Algorithm 2.2 gets worse exponentially in the dimensionality. However, Mulzer & Werner could further improve this result by generalizing the lifting argument as well as Lemma 2.13 by Miller and Sheehy.

Theorem 2.18 ([9, Theorem 1.7]). *Let $P \subset \mathbb{R}^d$ be a point set. Then a Tverberg partition of size $n/(4(d+1)^3)$ for P can be computed in $d^{\mathcal{O}(\log d)}|P|$ time.*

As in the case of Miller & Sheehy's algorithm, the Tverberg partitions returned by this algorithm are in general untolerated. Let us take again a look at Lemma 2.16. Although this is only a simplified version of the lemma used in Theorem 2.18, the discussed problem also affects the used generalization. Essential for the construction of the Tverberg partition in the proof of Lemma 2.16 was that c' is a Tverberg point for $\mathcal{T}'$. In the end of the last section, we have already argued that the removal of any point could lead to c' not being a Tverberg point anymore. Therefore, if any point from P is removed, the proof of the lemma does not guarantee a nonempty intersection of the convex hulls of the constructed Tverberg partition.

3 Tolerated Tverberg Partitions in Low Dimensions

3.1 One dimension

We start with the special case of tolerated Radon partitions in one dimension and later extend this solution to an arbitrary number of sets in the partition.

Problem 3.1 (Tolerated Radon Partitions in One Dimension)**.** Let P be a set of real numbers and t the tolerance parameter. The Problem is to compute a partition of P into two sets T_1 and T_2, so that that their convex hulls have a nonempty intersection even if up to t points are removed from P.

We assume P to contain at least $2t + 3$ points. We will see later that this involves no loss of generality. A simple solution to this problem is to sort some subset Q of size $2t + 3$ of P and then place the points in Q alternately in T_1 and T_2. All unused points in $Q \setminus P$ can be added to T_1 in order to obtain a partition of P. Figure 3.1 shows an example.

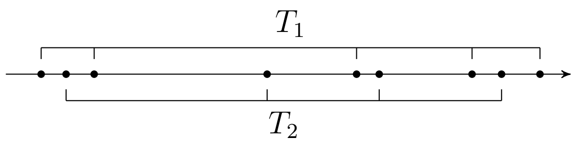

Figure 3.1: Tolerated Radon Partition with Tolerance 3

Proposition 3.2. *Given at least $2t + 3$ real numbers P, then Algorithm 3.1 computes a tolerated Radon partition $\{T_1, T_2\}$ with tolerance t in time $\mathcal{O}(|P| + t \log t)$. The deletion of up to t points R from P and the computation of a Radon point for $P \setminus R$ can be performed in $\mathcal{O}(t)$.*

Proof. Assume for the sake of contradiction the existence of a subset $R \subseteq P$ of size at most t whose deletion separates the convex hulls of T_1 and T_2. Let h be

Algorithm 3.1: 1d-Tolerated-Radon-Partition

```
input  : Set of real numbers $P$, tolerance parameter $t$
output : $t$-tolerated Radon partition of $P$
1 $Q \leftarrow$ any $2t+3$ points of $P$;
2 Sort $Q$. Let $q_1, q_2, \dots, q_{2t+3}$ be the sorted sequence;
3 $T_1 \leftarrow \{q_i \mid i \text{ is odd}\} \cup (P \setminus Q)$;
4 $T_2 \leftarrow \{q_i \mid i \text{ is even}\}$;
5 return $\{T_1, T_2\}$;
```

a separating hyperplane of $\mathsf{conv}(T_1 \setminus R)$ and $\mathsf{conv}(T_2 \setminus R)$. Define $T_1^- = T_1 \cap h^-$ and $T_1^+ = T_1 \cap h^+$ (T_2^-, T_2^+ are defined similarly). Since the size of P is odd, the following inequality always holds by construction of the partition:

$$|T_1^+| \geq |T_2^+| \tag{3.1}$$

$$|T_1^-| \geq |T_2^-| \tag{3.2}$$

Because h separates the resulting convex hulls, R has to contain either $T_1^+ \cup T_2^-$ or $T_1^- \cup T_2^+$. Using both inequalities and the fact that T_1 and T_2 are disjoint, we can get the desired contradiction by showing that R exceeds its maximum size t in both cases:

Case 1: $T_1^+ \cup T_2^- \subseteq R$

$$|T_1^+ \cup T_2^-| = |T_1^+| + |T_2^-| \overset{3.1}{\geq} |T_2^+| + |T_2^-| = |T_2|$$

Case 2: $T_1^- \cup T_2^+ \subseteq R$

$$|T_1^- \cup T_2^+| = |T_1^-| + |T_2^+| \overset{3.2}{\geq} |T_2^-| + |T_2^+| = |T_2|$$

Thus R has to be of size at least $|T_2|$. Since $|T_2| = t+1$, this is a contradiction to our initial assumption about the maximal size of R.

It remains to prove the running time in which we can delete t points R from P and compute a Radon point for $P \setminus R$. Suppose that each point $p \in P$ has an attached property `deleted`, that can be set to true to mark p as deleted. This makes the deletion rather simple: We just set for each point $p \in R$ the `deleted` property to true. To find a Radon point, we only need to return a number in the intersection of $\mathsf{conv}((T_1 \setminus R) \cap Q)$ and $\mathsf{conv}((T_2 \setminus R) \cap Q)$. By searching for the minimum and maximum number in $T_1 \setminus R$ and $T_2 \setminus R$, we can return a point in the intersection in linear time in the size of Q and t. Since $|Q| = 2t + 3$, this requires $\mathcal{O}(t)$ time. ■

We can further show that our assumption about the size of P involves no loss of generality as there exists no t-tolerated Radon partition for points sets of size strictly less than $2t + 3$.

Proposition 3.3. *Let $P \subset \mathbb{R}$ be a point set. There exists a t-tolerated Radon partition for P if and only if P is of size at least $2t + 3$.*

Proof. Let $P \subset \mathbb{R}^d$ be a set of size strictly less than $2t+3$ and $\{T_1, T_2\}$ a partition of P into two sets. Fix two points h_1 and h_2, such that $|P \cap (h_1, h_2)| = 1$. Let $T^-_{1,i}$ and $T^-_{2,i}$ denote the points in T_i to the left of h_1 and h_2, respectively. Define $T^+_{1,i}$ and $T^+_{2,i}$ in an analogue manner. The deletion of one of the four sets $T^-_{1,1} \cup T^+_{1,2}$, $T^+_{1,1} \cup T^-_{1,2}$, $T^-_{2,1} \cup T^+_{2,2}$, and $T^+_{2,1} \cup T^-_{2,2}$ separates the convex hulls of T_1 and T_2. We show that at least one of these sets is of size at most t.

Since $|T_1 \cup T_2| = |P| \leq 2t+2$, either $|T^-_{1,1} \cup T^+_{1,2}| \leq t+1$ or $|T^-_{1,1} \cup T^+_{1,2}| \leq t+1$ holds. Assume without loss of generality that the size of $T^-_{1,1} \cup T^+_{1,2}$ is less or equal to $t+1$. If $|T^-_{1,1} \cup T^+_{1,2}| \leq t$, we have found a subset of size at most t whose deletion separates the convex hulls of T_1 and T_2. Thus suppose that $|T^-_{1,1} \cup T^+_{1,2}| = t + 1$. Let $p^* \in P$ denote the point in between h_1 and h_2. If $p^* \in T^+_{1,2}$, then $T^+_{2,1} \cup T^-_{2,2}$ is a set of size t whose deletion separates the convex hulls of T_1 and T_2. On the other hand, if $p^* \in T^+_{1,1}$, then $T^+_{1,1} \cup T^-_{1,2}$ is a set of size at most t and the removal of this set separates the convex hulls of T_1 and T_2. Since the partition $\{T_1, T_2\}$ of P was arbitrary, the claim follows. ■

Our next goal is to extend algorithm 3.1 to the general case in which the partition can be of arbitrary size. The algorithm is based on the following observation: Given $2(t + 1)$ ordered numbers $Q = \{q_1, q_2, \ldots, q_{2(t+1)}\}$, the interval $[q_{t+1}, q_{t+2}]$ is contained in the convex hull $\mathsf{conv}(Q)$ even if up to t points are removed. We say Q covers the interval $[q_{t+1}, q_{t+2}]$ with tolerance t. The algorithm is divided into two steps: First, we bootstrap the tolerated Tverberg partition using Algorithm 3.1 to obtain a t-tolerated Radon partition $\{T_1, T_2\}$ for $2t + 3$ points of the initial point set. In a second step, we use the observation to find $m - 2$ disjoint point sets $T_3, T_4, \ldots, T_m$ that cover $\mathsf{conv}(T_1) \cup \mathsf{conv}(T_2)$ with tolerance t. Thus, after the deletion of any subset T of size t, $\mathsf{conv}(T_1 \setminus T) \cap \mathsf{conv}(T_2 \setminus T)$ is nonempty and is also contained in $\bigcap_{i=3}^{m} \mathsf{conv}(T_i \setminus T)$.

Before we go into more detail, let us take a look at an example in Figure 3.2. The first two elements of the partition form a 3-tolerated Radon partition consisting of 9 ($= 2 \cdot 3 + 3$) points. Both elements are covered with tolerance 3 by T_3 using additional 8 ($= 2(3 + 1)$) points.

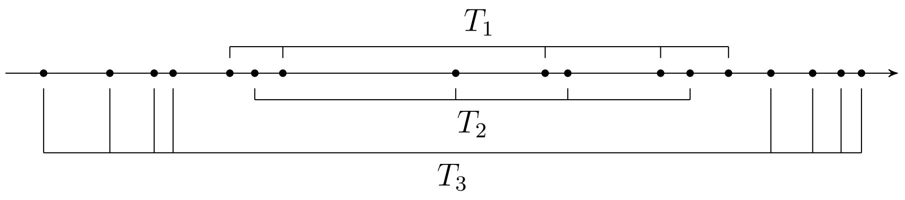

Figure 3.2: Tolerated Tverberg Partition of size 3 with tolerance 3

In general, our approach needs at least

$$\underbrace{2t + 3}_{t\text{-tolerated Radon partition}} + \underbrace{2(t + 1)(m - 2)}_{(m-2)\ t\text{-tolerated enveloping point sets}} = 2(t + 1)(m - 1) + 1$$

points to work. This condition is not particularly restrictive as it coincides with the bound by Soberón and Strausz [12].

Algorithm 3.2: $1d$-Tolerated-Tverberg-Partition

```
   input : Set of real numbers $P$, tolerance parameter $t$, size of partition $m$
   output: $t$-tolerated Tverberg partition of size $m$ of $P$
1  $l \leftarrow$ Select $(P, (m-2)(t+1))$;
2  $h \leftarrow$ Select $(P, |P| - (m-2)(t+1) - 1)$;
3  LowPoints $\leftarrow \{p \in P \mid p \leq l\}$;
4  MiddlePoints $\leftarrow \{p \in P \mid l < p < h\}$;
5  HighPoints $\leftarrow \{p \in P \mid h \leq p\}$;
6  $\{T_1, T_2\} \leftarrow$ 1d-Tolerated-Radon-Partition (MiddlePoints, $t$);
7  foreach $i \in \{3, 4, \ldots, m\}$ do
8      $L \leftarrow$ remove any $t + 1$ points in LowPoints;
9      $H \leftarrow$ remove any $t + 1$ points in HighPoints;
10     $T_i \leftarrow L \cup H$;
11 return $\{T_1, T_2, \ldots, T_m\}$;
```

Algorithm 3.2 is based on the strategy discussed in the last paragraph. Lines 1–5 partition the point set into three parts: Low and high points that will be used to construct the enveloping elements of the partition and middle points that will be partitioned using the algorithm for tolerated Radon partitions. We need $2t + 3$ points for the middle part of the partition to apply Algorithm 3.1 and at least $(m - 2)(t + 1)$ points in each other part, low and high points, in order to use the observation. We use the `Select` procedure to find the $(m - 2)(t + 1)$-lowest and the $(m - 2)(t + 1)$-highest point in P and then split the point set according to the relative order to those two points. After computing the t-tolerated Radon partition for the middle points in line 6, we build the outer parts of the Tverberg partition in lines 7–10.

Proposition 3.4. *Given a set of real numbers P of size at least $2(t + 1)(m - 1) + 1$, Algorithm 3.2 computes a t-tolerated Tverberg partition of size m in time $\mathcal{O}(|P| + t \log t)$. The deletion of up to t points R from P and the computation of a Tverberg point for $P \setminus R$ can be performed in $\mathcal{O}(t)$.*

Proof. We have already discussed the correctness of this algorithm. We will now analyze the complexity of the construction and the running time in which we can compute a Tverberg point of the constructed Tverberg partition after up to t points were removed. We start with the construction time.

The `Select` procedure requires time $\mathcal{O}(|P|)$ if implemented as described in the book by Cormen et. al. [4, Chapter 9]. The initial partition in lines 3–5 can be found in one iteration over P and therefore in time $\mathcal{O}(|P|)$. The computation of the tolerated Radon partition takes $\mathcal{O}(|P| + t \log t)$ time. The construction of the outer partitions in line 7–10 does not require the picking of the points to happen in a particular order, so the usage of a simple data structure like a list gives a running time of $\mathcal{O}(|P|)$ for this step. All in all, the complete running time for Algorithm 3.2 is $\mathcal{O}(|P| + t \log t)$ as claimed.

The deletion can be carried out in $\mathcal{O}(t)$ time in the same way as described in the proof of Proposition 3.2 by setting the `deletion` property to true. Let us now take a look at the partition after the deletion of up to t points. By construction of the enveloping sets, we know they still cover the whole interval of both inner sets. Thus it is sufficient to compute an intersection of T_1 and T_2. As proved in Proposition 3.2, this can be performed in $\mathcal{O}(t)$. ■

Interestingly, the generalization to tolerated Tverberg partitions has no effect on the running time. However, Algorithm 3.1 can be generalized in a more direct way that improves the size of the computed tolerated Tverberg partition. As with the Radon partitions, we sort the points and then place them alternately in each partition. Figure 3.3 shows an example. To understand why this approach still works, it helps to think of building cells with

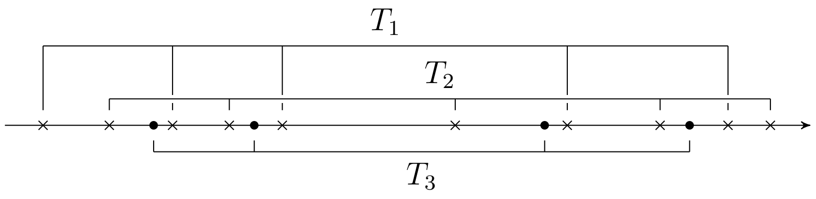

Figure 3.3: Tolerated Tverberg Partition of size 3 with tolerance 3

the first $m-1$ sets of the partition. Each two neighboring cells of one set in the partition have a border in common, consisting of exactly one point. For example, the first and the fourth point in Figure 3.3 form a cell of T_1. The fourth and the 7th point form another cell. The last element of the Partition, T_m, is special: It is not used to build cells, but it fills the cells of the other elements. The ith point of T_m together with the ith cells of the other sets form an untolerated Tverberg partition of size m. The idea is now to build $t+1$ cells, such that after the deletion of up to t points at least one is still intact.

Before we take a look at the complete algorithm and formally prove the correctness, let us reason about the minimum number of points needed to use this approach. Each of the first $m-1$ sets of the partition has to consist of at least $t+1$ cells, where two neighboring cells have a point in common. The last element of the partition is used to fill the $t+1$ cells of each other set with at least one point. This results in the following lower bound on the size of the point set

$$\underbrace{(m-1)(t+2)}_{\text{first } m-1 \text{ sets}} + \underbrace{t+1}_{m\text{th set}} = m(t+2)-1$$

Algorithm 3.3: $1d$-Tolerated-Tverberg-Cells

```
  input  : Set of real numbers $P$, tolerance parameter $t$, size of partition $m$
  output : $t$-tolerated Tverberg partition of size $m$ of $P$
1 $Q \leftarrow$ any $m(t+2)-1$ points of $P$;
2 Sort $Q$. Let $q_1, q_2, \dots, q_{m(t+2)-1}$ be the sorted sequence;
3 foreach $i \in \{1, 2, \dots, m\}$ do
4     $Q_i \leftarrow \{q_j \in Q \mid j \mod m = i-1\}$
5 $T_1 \leftarrow T_1 \cup (P \setminus Q)$;
6 return $\{T_1, T_2, \dots, T_m\}$
```

Algorithm 3.3 shows the steps in more detail. In lines 1–2, a subset $Q \subseteq P$ of suitable size is selected and then sorted. The points within Q are alternately placed in the sets of the partition in line 4. Finally, any remaining points are placed in the first part of the partition in line 5.

Proposition 3.5. *Given a set of real numbers P of size at least $m(t+2)-1$, Algorithm 3.3 computes a t-tolerated Tverberg partition for P of size m in time $\mathcal{O}(|P| + mt \log mt)$. The deletion of up to t points R from P and the computation of a Tverberg point for $P \setminus R$ can be performed in $\mathcal{O}(m+t)$.*

Proof. Let $T_1, T_2, \ldots, T_m$ be the output obtained by Algorithm 3.3 and $R \subseteq P$ of size at most t. We prove the following two assertions. The statement then follows by Helly's theorem.

(α) $\forall i \in \{1, 2, \ldots, m-1\} : \mathsf{conv}(T_i \setminus R) \cap \mathsf{conv}(T_m \setminus R) \neq \emptyset$

(β) $\forall i, j \in \{1, 2, \ldots, m-1\}, i < j : \mathsf{conv}(T_i \setminus R) \cap \mathsf{conv}(T_j \setminus R) \neq \emptyset$

(α) Consider the output of Algorithm 3.1 on the input $T_i \cup T_m$ ($i \in \{1, 2, \ldots, m-1\}$) and tolerance parameter t. It is identical to the partition $\{T_i, T_m\}$. Since this is a t-tolerated Radon partition, the assertion follows.

(β) The proof is similar to the proof of Proposition 3.2. Assume for the sake of contradiction that there exist two elements T_i, T_j ($i, j \in \{1, 2, \ldots, m-1\}$) of the constructed Tverberg partition whose convex hulls have an empty intersection if R is removed from P. Let h be a separating hyperplane of $\mathsf{conv}(T_i \setminus R)$ and $\mathsf{conv}(T_j \setminus R)$. Define k to be $|T_m \cap h^-|$ (i.e. the number of cells to the left of h). We know by construction of the partition

$$\forall i \in \{1, 2, \ldots, m-1\} : |T_i \cap h^-| \in \{k, k+1\}$$

Since R separates the convex hulls of both elements, R has to contain either $(T_i \cap h^-) \cup (T_j \cap h^+)$ or $(T_j \cap h^-) \cup (T_i \cap h^+)$. We obtain a contradiction by showing that R exceeds in both cases its maximal size.

$$|(T_i \cap h^-) \cup (T_j \cap h^+)| \geq k + |T_j| - (k+1) = |T_j| - 1 = t+1$$
$$|(T_j \cap h^-) \cup (T_i \cap h^+)| \geq k + |T_i| - (k+1) = |T_i| - 1 = t+1$$

We now analyze the running time of the algorithm. In order to construct the partition we have to sort a subset of P of size $m(t+2)-1$. This can be done in time $\mathcal{O}(mt \log mt)$. If P is greater than this subset, the remaining points are added to the first element of the partition. This requires $\mathcal{O}(|P|)$ time and thus $\mathcal{O}(|P| + mt \log mt)$ in total.

To compute a Tverberg point of depth m after the deletion of up to t points R, we have to compute the convex hulls of $T_1 \setminus R, T_2 \setminus R, \dots, T_m \setminus R$ and find a point in the intersection. The point removal itself is again done by setting the `deletion` attributes of the points in R to true in $\mathcal{O}(t)$ time. Since all sets $T_1, T_2, \dots, T_m$ are sorted, the minimum and maximum of each T_i can be found in constant time. To compute the minimum and maximum of each set $T_i \setminus R$ ($i \in \{1, 2, \dots, m\}$), we have to skip over the deleted points. Thus, we can find a Tverberg point after the removal of R in $\mathcal{O}(m + t)$ time. ■

Note that the points between two consecutive points of T_m can be assigned to the other elements of the partition in any order. The proof (especially assertion (β)) remains still valid as long as there is a point of each set T_i ($i \in \{1, 2, \dots, m-1\}$) between each two consecutive points of T_m. However, Algorithm 3.3 assigns them in a sorted order. We can improve the running time by not sorting the whole point set. Instead, we compute T_m using the `Select` procedure by searching each mth point and then partition the rest of P into $t+2$ intervals $(-\infty, p_m), (p_m, p_{2m}), \dots, (p_{(t+1)m}, \infty)$. The first $m-1$ sets of the partition can then be easily constructed by selecting one point in each of those intervals.

A naive algorithm that selects each mth point of P and then partitions $P \setminus T_m$ into intervals would have a running time of

$$\underbrace{\mathcal{O}(m \cdot |P|)}_{\text{computation of } T_m} + \underbrace{\mathcal{O}(\log(t) \cdot |P|)}_{\text{partitioning of } P \setminus T_m} = \mathcal{O}(m^2 t + mt \log(t)) \quad \text{if } |P| = m(t+2) - 1$$

To actually improve the running time, we have to be more careful. The optimized algorithm is based on a quicksort-like divide & conquer strategy. In each step, we select the median of T_m and split P into two sets P^- and P^+ that contain all points lower and higher than the median, respectively. We recursively search for the other elements of T_m in P^- and T_+. If there is only one point of T_m left in the search interval, we know due to the splitting in points higher and lower than the median, that all other points lie between two consecutive points of T_m. The splitting thus serves two purposes: Improving the search time of T_m and on-the-fly partitioning of $P \setminus T_m$ into intervals $(-\infty, p_m), (p_m, p_{2m}), \dots, (p_{(t+1)m}, \infty)$.

The complete steps are shown in Algorithm 3.4. As in the algorithms before, we choose a subset Q of suitable size if P is too big. In lines 5–9, we search the median of T_m and partition Q according to the value of the median. If there are still points of T_m in Q missing, we recursively apply the algorithm on Q^- and Q^+ and combine the results in lines 13–16. Lines 17–20 cover the base case: The set $Q^* \in \{Q^+, Q-\}$ contains all points from Q in the interval $(p_{(k-1)m}, p_{km})$ or $(p_{km}, p_{(k+1)m})$. These points are then distributed among the first $m-1$ elements of the constructed tolerated Tverberg partition to ensure, that their intersection with the interval is nonempty.

Algorithm 3.4: 1d-Tolerated-Tverberg-Cells*

```
   input  : Set of real numbers $P$, tolerance parameter $t$, size of partition $m$
   output : $t$-tolerated Tverberg partition for $P$ of size $m$
 1 if $|P| > m(t+2)-1$ then
 2     $Q \leftarrow$ any $m(t+2)-1$ points of $P$;
 3 else
 4     $Q \leftarrow P$;
 5 $k \leftarrow \lceil(|Q| \text{ div } m)/2\rceil$;
 6 $q_{km} \leftarrow$ Select $(Q, km)$;
 7 $T_m \leftarrow \{q_{km}\}$;
 8 $Q^- \leftarrow \{q \in Q \mid q < q_{km}\}$;
 9 $Q^+ \leftarrow \{q \in Q \mid q > q_{km}\}$;
10 foreach $i \in \{1, 2, \ldots, m-1\}$ do
11     $T_i \leftarrow \emptyset$;
12 foreach $Q^* \in \{Q^-, Q^+\}$ do
13     if $|Q^*| \geq m$ then
14         $\{T_1^*, T_2^*, \ldots, T_m^*\} \leftarrow$ 1d-Tolerated-Tverberg-Cells*$(Q^*, t, m)$;
15         foreach $i \in \{1, 2, \ldots, m\}$ do
16             $T_i \leftarrow T_i \cup T_i^*$;
17     else
18         foreach $i \in \{1, 2, \ldots, m-1\}$ do
19             $q^* \leftarrow$ remove any point from $Q^*$;
20             $T_i \leftarrow T_i \cup \{q^*\}$;
21 $T_1 \leftarrow P \setminus Q$;
22 return $\{T_1, T_2, \ldots, T_m\}$;
```

Theorem 3.6. *Given a set of real numbers P of size at least $m(t+2)-1$, Algorithm 3.4 computes a t-tolerated Tverberg partition for P of size m in time $\mathcal{O}(|P| + mt\log t)$. After the deletion of up to t points from P, a Tverberg point can be determined within $\mathcal{O}(mt)$ time.*

Proof. Let P be a point set of size exactly $m(t+2)-1$. We can safely ignore the case in which P is bigger than $m(t+2)-1$, as the extension from a t-tolerated partition of size m for a subset Q of P to a t-tolerated partition of size m for the whole point set P in Algorithm 3.4 is trivially right. Let $T_1, T_2, \ldots, T_m$ be the output of Algorithm 3.4 on input P, tolerance parameter t and desired size m. To reuse the proof of Proposition 3.5, we have to show the following two assertions

(α) T_m contains each mth point of P

(β) Let $p_m, p_{2m}, \ldots, p_{(t+1)m}$ be the sorted sequence of T_m.
Then $\forall i \in \{1, 2, \ldots, m-1\}$:

$$T_i \cap (-\infty, p_m) \neq \emptyset \wedge T_i \cap (p_m, p_{2m}) \neq \emptyset \wedge \ldots \wedge T_i \cap (p_{(t+1)m}, \infty) \neq \emptyset$$

(α) Intuitively, it is clear that the algorithm always selects an element whose rank is a multiple of m and adds this to T_m. This is true even in the recursive invocations, since we split the point set only at elements of T_m. The formal proof is a bit technical and can safely be skipped.

We first prove that the rank in the original point set of each selected element of T_m in line 6 is indeed a multiple of m. We show this by induction over the recursion depth n of the algorithm. Let $P^{(i)}$ be some input set to an instance of the algorithm in the ith level of the recursion tree and let $f : \mathscr{P}_F(\mathbb{R}) \times \mathbb{R} \to \mathbb{N}$ be a function that returns the rank of an element with respect to some set, where $\mathscr{P}_F(\mathbb{R})$ denotes all finite subsets of $\mathbb{R}$.

Inductive Hypothesis

$$H(n) \Leftrightarrow \forall p_i \in P^{(n)} : f\left(P^{(n)}, p_i\right) \equiv f\left(P, p_i\right) \mod m$$

Base Case $n = 1$

Since $P^{(1)} = P$, $H(1)$ obviously holds.

Inductive Step $n-1 \to n$

If the recursive call in line 19 was invoked with Q^- as its argument, the inductive hypothesis implies that

$$\forall p_i \in P^{(n)} : f\left(P^{(n)}, p_i\right) \equiv f\left(Q^-, p_i\right) \mod m \qquad (3.3)$$

Since Q^- contains all points in P lower than p_{km}, the ranks in Q^-

have not changed:

$$\forall p_i \in Q^- : f\left(Q^-, p_i\right) = f\left(P, p_i\right) \tag{3.4}$$

Equations 3.3 and 3.4 imply $H(n)$.

Suppose now the recursive call in line 19 was invoked with Q^+ as its argument. Again, we can apply the inductive hypothesis:

$$\forall p_i \in P^{(n)} : f\left(P^{(n)}, p_i\right) \equiv f\left(Q^+, p_i\right) \mod m \tag{3.5}$$

The set Q^+ contains all points in P that are greater than p_{km}. Thus, the ranks of the points in Q_i are decreased by km compared to their ranks in P.

$$\forall p_i \in Q^+ : f\left(Q^+, p_i\right) = f\left(P, p_i\right) - km \equiv f(P, p_i) \mod m \tag{3.6}$$

Equations 3.5 and 3.6 imply $H(n)$.

Since $H(n)$ holds in both possible cases, this concludes the induction.

So far, we have proved that the selection of points for T_m is right. It remains to show that all elements whose rank is a multiple of m are found. In each step of the algorithm, one point of T_m is found and removed from the current point set (i.e., is no part of $Q^- \cup Q^+$). In lines 13–16, the search continues in the remaining point set, so no possible candidate for T_m is removed. The algorithm terminates if $|P^{(n)}| < 2m$. As we have already proved, the ranks of the points in $P^{(n)}$ are equivalent to the ranks in the original point set modulo m. Therefore, $P^{(n)}$ contains only one point of the original point set whose rank is a multiple of m. This point is found in line 6 and correctly added to T_m. Because there are no more candidates in the point set, the termination is correct.

(β) Similar to case (α), we will prove the statement by induction on the recursion depth.

Inductive Hypothesis Let $T_1, T_2, \ldots, T_m$ be the output of Algorithm 3.4 on input $P^{(n)}$ and let $p_1^{(m)}, p_2^{(m)}, \ldots, p_{|T_m|}^{(m)}$ be the sorted sequence of T_m. Then $H(n) \Leftrightarrow \forall I \in \{(-\infty, p_1^{(m)}), (p_1^{(m)}, p_2^{(m)}), \ldots, (p_{|T_m|}^{(m)}, \infty)\}$: If $|I \cap P^{(n)}| \geq m - 1$, then each element T_i ($i \in \{1, 2, \ldots, m-1\}$) contains a unique point of $I \cap P^{(n)}$.

Base Case $n = 1$

Because the recursion depth is only 1, P has to be of size strictly less than $2m$. Thus, T_m contains only one point p_m. The sets Q^- and Q^+ then contain all points in $P \cap (-\infty, p_m)$ and $P \cap (p_m, \infty)$, respectively. Since $|P^{(1)}| < 2m$, we know that $|Q^-|, |Q^+| < m$. If $|Q^-| \geq m - 1$, then a point from $Q^- = (-\infty, p_m) \cap P$ is added to each set $T_1, T_2, \ldots, T_{m-1}$ in lines 17–20. The same holds for Q^+. This implies $H(1)$.

Inductive Step $n - 1 \to n$

Let $T_1, T_2, \ldots, T_m$ be the output of Algorithm 3.4 on input $P^{(n)}$ and let $I \in \{(-\infty, p_1^{(m)}), (p_1^{(m)}, p_2^{(m)}), \ldots, (p_{|T_m|}^{(m)}, \infty)\}$ be some interval induced by T_m that contains at least $m - 1$ points of $P^{(n)}$. Because $P^{(n)}$ is divided at a point $p_{km} \in T_m$ into the sets Q^- and Q^+, all points of $P^{(n)}$ in I are either in Q^- or Q^+, but not distributed between both. Without loss of generality, let $I \cap P^{(n)}$ be a subset of Q^-. If $|Q^-| < m$, then $Q^- = I \cap P^{(n)}$ holds because $I \cap P^{(n)}$ is of size at least $m - 1$. In this case, a point in this interval is assigned to each element T_i $(i \in \{1, 2, \ldots, m - 1\})$ of the partition in lines 17–20. If $|Q^-| \geq m$, then the algorithm is recursively invoked with Q^- as its input in line 14. Let $T_1^*, T_2^*, \ldots, T_m^*$ be the result of this recursive invocation. Since the recursion depth of this invocation is less than n, we can apply the inductive hypothesis. Thus, each set T_i^* $(i \in \{1, 2, \ldots, m-1\})$ contains at least one point of Q^- in each interval $(-\infty, q_1^{(m)}), (q_1^{(m)}, q_2^{(m)}), \ldots, (q_{|T_m^*|}^{(m)}, +\infty)$, where $q_1^{(m)}, q_2^{(m)}, \ldots, q_{|T_m^*|}^{(m)}$ is the sorted sequence of points in T_m^*. Since T_m^* contains all points of T_m lower than p_{km}, one of the intervals $(-\infty, q_1^{(m)}), (q_1^{(m)}, q_2^{(m)}), \ldots, (q_{|T_m^*|}^{(m)}, +\infty)$ coincides with I. Thus, each set T_i^* $(i \in \{1, 2, \ldots, m - 1\}$ contains at least one point of $Q^- \cap I = I \cap P^{(n)}$. Because each set T_i^* $(i \in \{1, 2, \ldots, m - 1\})$ is contained in the corresponding set T_i of the returned partition, we know that T_i also contains a point of $P^{(n)}$ in I, which concludes the proof.

Statement (α) asserts, that T_m contains each mth point of the input point set. Thus, each interval $\{(-\infty, p_1^{(m)}), (p_1^{(m)}, p_2^{(m)}), \ldots, (p_{|T_m|}^{(m)}, \infty)\}$ contains exactly $m - 1$ points. Together with the induction, this implies (β).

So far we have proved, that the convex hulls of the elements of the returned sets $T_1, T_2, \ldots, T_m$ cannot be separated by the removal of up to t points. We still have to check that these sets form a partition of P. Each point that is assigned to an element T_i is directly removed and will not be assigned to any other element again. Also, points will only be removed if they are assigned. Together, this implies that the result is a partition of the input set.

The search for the splitting element q_{km} in each step can be performed in $\mathcal{O}(mt)$ time. Everything else except the recursive invocation can also be computed within $\mathcal{O}(mt)$ time. Since q_{km} is the median of the input set, the input sets of the recursive calls is less than $|P|/2$. The algorithm terminates if the input set is of size strictly less than $2m$. This leads to the following recursion inequality for the total running time

$$\begin{aligned}&\forall n \geq 2m : T(n) \leq 2T(n/2) + \mathcal{O}(n)\\ &\forall n < 2m : T(n) \leq \mathcal{O}(n)\end{aligned}$$

which resolves to $\mathcal{O}(mt \log t)$ as claimed. ■

The minimum number of points needed to apply Algorithm 3.3 and 3.4 is different to the bound by Soberón and Strausz. We will now focus on the relationship between both bounds of the tolerated Tverberg problem. Surprisingly, we can show that both preceding algorithms give a great improvement compared to the bound by Soberón and Strausz for the one dimensional case. We first show that the new bound is always less or equal than the Soberón-Strausz bound for all values of m and t.

$$\begin{aligned} m(t+2)-1 &\overset{!}{\leq} 2(t+1)(m-1)+1 && \Leftrightarrow\\ (m-1)(t+2)+t+1 &\overset{!}{\leq} 2(t+1)(m-1)+1 && \Leftrightarrow\\ t+1 &\overset{!}{\leq} t(m-1)+1 && \Leftrightarrow\\ 0 &\overset{!}{\leq} t(m-2) \end{aligned}$$

The last inequality holds since $m \geq 2$ is a precondition. The inequality also implies, that the new bound is strictly better if $t > 0$ or $m > 2$. That is, the new bound is better except in the special cases of untolerated Tverberg partitions or tolerated Radon partitions.

We now analyze the asymptotic behaviour of both bounds for large values of

m and t.

$$\begin{aligned}
&\lim_{(m,t)\to(\infty,\infty)} \frac{m(t+2)-1}{2(t+1)(m-1)+1} \\
=&\lim_{(m,t)\to(\infty,\infty)} \frac{1}{2+\frac{1}{(t+1)(m-1)}} + \frac{1}{2(t+1)} + \frac{1}{2(m-1)} \\
=&\frac{1}{2}
\end{aligned}$$

Thus, for large values of m and t only about half of the points compared to the bound by Soberón and Strausz are required. We can show that the new bound cannot be improved.

Theorem 3.7. *Let P be a set of a real numbers. There exists a t-tolerated Tverberg partition for P of size m if and only if P is of size at least $m(t+2)-1$.*

Proof. We have already proved in Proposition 3.5 the existence of a t-tolerated Tverberg partition of size m for point sets having the required amount of elements.

Let P' be a set of real numbers of size strictly less than $m(t+2)-1$. Assume there is a t-tolerated Tverberg partition $P'_1, P'_2, \ldots, P'_m$ of size m for P'. By the averaging argument, there is a set P'_i of size less or equal to $t+1$. On the other hand, P'_i has to be of size at least $t+1$ since the partition is t-tolerated. Therefore, P is of size exactly $t+1$. Suppose there exists another set P'_j such that $|P'_i \cup P'_j| < 2t+3$. Because P'_i and P'_j are part of a t-tolerated Tverberg partition, $\{P'_i, P'_j\}$ forms a t-tolerated Radon partition for $P'_i \cup P'_j$. This contradicts Proposition 3.3.

Suppose now there exists no set P'_j such that $|P'_i \cup P'_j| < 2t+3$ holds. Then each other set P'_j has to be of size at least $2t+3-|P'_i|$. We can use this to lower bound the size of P'.

$$\begin{aligned}
m(t+2)-1 > |P'| &= |P'_i| + |\textstyle\bigcup_{j\in\{1,2,\ldots,m\}, j\neq i} P'_j| \\
&\geq |P'_i| + (m-1)(2t+3-|P'_i|) \\
&\geq (t+1) + (m-1)(t+2) = m(t+2)-1 \qquad ↯
\end{aligned}$$

We have obtained contradictions in both possible cases, which proves the theorem. ■

3.2 Multiple Dimensions

We use a simple dimension reduction similar to the lifting argument (Lemma 2.16) by Mulzer and Werner [9] that enables us to apply the algorithms in the last section to multi-dimensional input. Given a point set $P \subseteq \mathbb{R}^d$, let h be a hyperplane that splits P evenly. We then partition P into pairs $T_i = \{p_i^-, p_i^+\}$, where $p_i^- \in h^-$ and $p_i^+ \in h^+$. Each of those pairs is projected onto the intersection of the connecting line segment $\overline{p_i^+ p_i^-}$ and h. We thus obtain $|P|/2$ points of dimension $d-1$. Figure 3.4 displays an example. Pairs are indicated with dashed lines, crosses mark the points onto which the pairs are projected.

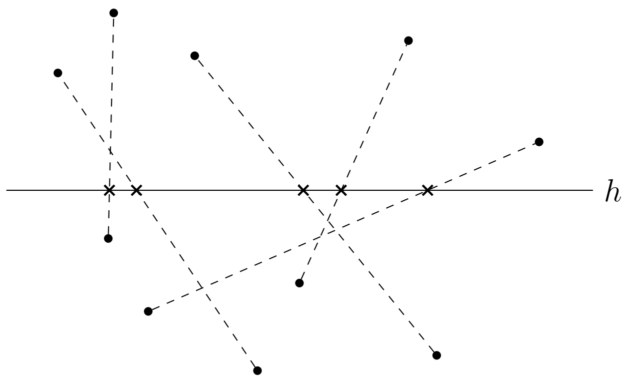

Figure 3.4: Dimension Reduction

Let $p_i' \in \mathbb{R}^{d-1}$ be the projection of T_i and $\mathcal{T}' = \{T_1', T_2', \ldots, T_m'\}$ a t-tolerated Tverberg partition of $P' = \{p_1', \ldots, p_{|P|/2}'\}$. It is easy to check that we obtain a Tverberg partition $\mathcal{T}$ of size m and tolerance t for P by replacing each q_i in $\mathcal{T}'$ by its corresponding pair p_i^-, p_i^+. The algorithm is now clear: We reduce the dimensionality of the initial point set P until we can apply Algorithm 3.4 and then lift the solution for one dimension to the original dimensionality.

Algorithm 3.5 follows this approach. If the dimensionality is one, the algorithms just uses our result from the last section (lines 1–2). The hyperplane h is computed in line 4. In lines 5–8, the points are projected onto h. Finally, the result for $d-1$ dimensions is lifted to the original dimensionality (lines 11–12).

Proposition 3.8. *Given a set $P \subset \mathbb{R}^d$ of size at least $2^{d-1}(m(t+2)-1)$, Algorithm 3.5 computes a Tverberg partition of size m and tolerance t for P in time $\mathcal{O}(|P| + mt \log t)$.*

Proof. We have already proved the correctness of the algorithm. It still remains

Algorithm 3.5: DimReduct-Tolerated-Tverberg

```
input  : point set $P \subset \mathbb{R}^d$, tolerance parameter $t$, size of partition $m$
output : $t$-tolerated Tverberg partition for P of size $m$
1  if $d = 1$ then
2      return 1d-Tolerated-Tverberg-Cells*($P$,$m$,$t$)
3  $h \leftarrow$ hyperplane s.t. $|P \cap h^-| = |P \cap h^+|$;
4  Choose a basis $\mathcal{B}$ of the affine subspace $h$;
5  foreach $i \in \{1, 2, \dots, |P \cap h^-|\}$ do
6      $p_i^- \leftarrow$ remove any point from $P$ that is an element of $P \cap h^-$;
7      $p_i^+ \leftarrow$ remove any point from $P$ that is an element of $P \cap h^+$;
8      $q_i \leftarrow$ coordinates of $\overline{p_i^- p_i^+} \cap h$ with respect to $\mathcal{B}$;
9  $Q \leftarrow \{q_1, q_2, \dots, q_{|P \cap h^-|}\}$;
10 $\{T'_1, T'_2, \dots, T'_m\} \leftarrow$ DimReduct-Tolerated-Tverberg($Q$,$m$,$t$);
11 foreach $j \in \{1, 2, \dots, m\}$ do
12     $T_j \leftarrow \{p_i^-, p_i^+ \mid q_i \in T'_j\}$;
13 return $\{T_1, T_2, \dots, T_m\}$;
```

for us to show the running time and to prove the lower bound on the size of P.

We can compute the hyperplane h by searching a median for P projected onto the first dimension. Using the `Select` procedure, this takes time $\mathcal{O}(|P|)$. The dimension reduction as well as the lifting requires also linear time in the size of P. We obtain the following recurrence relations for appropriate constants c_1, c_2

$$\begin{aligned} T(n,d) &= T(n/2, d-1) + c_1 n \\ T(n,1) &= c_2(mt\log t + n) \end{aligned}$$

which resolves to

$$\begin{aligned} T(n,d) &= c_2\left(mt\log t + \frac{n}{2^{d-1}}\right) + \sum_{j=0}^{d-2} c_1 \frac{n}{2^j} \\ &= c_2\left(mt\log t + \frac{n}{2^{d-1}}\right) + c_1 n\left(2 - \frac{1}{2}^d\right) \in \mathcal{O}(mt\log t + n) \end{aligned}$$

This proves the claimed running time. Note that the parameter d is hidden in the size n of P. We continue with proving the bound on the size of P. In each step, we project two points in P onto one point in h, thus halving the size of the point set. To apply Algorithm 3.4, we need at least $m(t+2)-1$ one-dimensional points. Therefore, it is sufficient if P is of size at least $2^{d-1}(m(t+2)-1)$. ■

Again, we are interested in a comparison between the bound of Algorithm 3.5 and the Soberón-Strausz-bound.

$$\begin{aligned} 2^{d-1}(m(t+2)-1) &\overset{!}{<} (d+1)(m-1)(t+1)+1 && \Leftrightarrow \\ 0 &\overset{!}{<} \Big((d+1)-2^{d-1}\Big)mt + \Big((d+1)-2^{d}\Big)m \\ &\quad + 2^{d-1} - (d+1)t - d \end{aligned}$$

In general, this is false as the right hand side of the inequality becomes negative if $d \geq 3$. Let us examine the conditions under which the new bound is better for the two-dimensional case.

$$\begin{aligned} 0 &\overset{!}{<} mt - m - 3t \\ \frac{m}{m-3} &\overset{!}{<} t \end{aligned}$$

This holds for instance if $t \geq 5$ or $m \geq 7 \wedge t \geq 2$. Thus, Algorithm 3.5 can construct a t-tolerated Tverberg partition of size m in $\mathbb{R}^2$ using less points than the Soberón-Strausz bound implies for almost all values of m and t.

There is a slight optimization to the presented dimension reduction: By choosing a hyperplane that splits P into two even sets *and* is spanned by d points in P, the points in $P \cap h$ can be projected onto the $d-1$ dimensional subspace without having to sacrifice another point. Unfortunately, this optimization is too weak to derive a better bound for special cases in dimensions higher than 2.

4 Approximation Preserving Reductions

The focus of this chapter lies in the development of approximation preserving reductions of the tolerated Tverberg problem to the untolerated version, as those let us reuse existing approximating algorithms for the untolerated Tverberg problem to compute tolerated Tverberg partitions. An *approximation preserving reduction* is a reduction of a problem Π to a problem Π' that leads for every α-approximation algorithm for Π to an $f(\alpha)$-approximation algorithm for Π', where f is some function. For a more thorough discussion about this kind of reductions, we refer the reader to the book by Williamson and Shmoys [19].

4.1 Reduction to the Untolerated Tverberg Problem

We present three different approximation preserving reductions to the untolerated Tverberg problem. All reductions can be applied to any approximation algorithm for the untolerated Tverberg problem.

4.1.1 Point Stabilization

In general, the output of any approximation algorithm $\mathcal{A}_{UT}$ for the untolerated Tverberg problem is of tolerance 0, as each point could be essential for the intersection of the convex hulls of the partition. One way to overcome this problem is to compute points with depth of at least $t+1$ for disjoint subsets of P. These points are still contained in the convex hulls of their corresponding subsets after the removal of up to t points. In this sense, they are stable. The disjoint subsets can be used to construct a t-tolerated Tverberg partition for P out of any untolerated Tverberg partition for the stabilized points.

Let $\mathcal{A}_{UT}$ and $\mathcal{A}_{CP}$ be approximation algorithms for the untolerated Tverberg problem and the centerpoint problem, respectively. Assume that $\mathcal{A}_{UT}$ returns a Tverberg partition of size $m_{UT}(|P|)$ for any point set P and $\mathcal{A}_{CP}$ returns a point of depth $t+1$ with respect to an input point set of size at least $s_{CP}(t+1)$. In Algorithm 4.1, we partition the point set P into $n' = \lfloor |P|/s_{CP}(t+1) \rfloor$ sets $Q_1, Q_2, \ldots, Q_{n'}$, each of size $s_{CP}(t+1)$, and use algorithm $\mathcal{A}_{CP}$ to obtain for each set a point p'_i of depth $t+1$ (lines 1–4). Applying algorithm $\mathcal{A}_{UT}$ in line 5, we obtain a Tverberg partition $\mathcal{T}' = \{T'_1, T'_2, \ldots, T'_{m_{UT}(n')}\}$ of the stabilized point set $P' = \{p'_1, p'_2, \ldots, p'_{n'}\}$. In lines 6–7, each point p'_i in the elements of T' is replaced by its corresponding subset Q_i. The so modified partition is returned as result of the algorithm.

Algorithm 4.1: Point-Stabilization

```
  input : Point set $P \subset \mathbb{R}^d$, tolerance parameter $t$, algorithm $\mathcal{A}_{CP}$,
          algorithm $\mathcal{A}_{UT}$
1 $n' \leftarrow \lfloor |P|/s_{CP}(t+1) \rfloor$;
2 Partition $P$ into $n'$ sets $Q_1, Q_2, \ldots, Q_{n'}$, each of size at least $s_{CP}(t+1)$;
3 foreach $i \in \{1, 2, \ldots, n'\}$ do
4     $p'_i \leftarrow \mathcal{A}_{CP}(Q_i)$;
5 $\{T'_1, T'_2, \ldots, T'_{m_{UT}(n')}\} \leftarrow \mathcal{A}_{UT}(\{p'_i \mid i \in \{1, 2, \ldots, n'\}\})$;
6 foreach $i \in \{1, 2, \ldots, m_{UT}(n')\}$ do
7     $T_i \leftarrow \bigcup_{p'_j \in T'_i} Q_j$;
8 return $\{T_1, T_2, \ldots, T_{m_{UT}(n')}\}$
```

Theorem 4.1. *Let $\mathcal{A}_{UT}$, $\mathcal{A}_{CP}$ be as above and T_{UT}, T_{CP} the respective running time functions. Algorithm 4.1 constructs a t-tolerated Tverberg partition of size $m_{UT}(\lfloor |P|/s_{CP}(t+1) \rfloor)$ for any input point set $P \subset \mathbb{R}^d$ in time $\mathcal{O}(T_{CP}(s_{CP}(t+1)) \cdot |P|/s_{CP}(t+1) + T_{UT}(|P|/s_{CP}(t+1)))$.*

Proof. We start with proving the correctness of Algorithm 4.1 and then show the running time.

As $\mathcal{T}' = \{T'_1, T'_2, \ldots, T'_{m_{UT}(n')}\}$ is a Tverberg partition of P', we know by definition that

$$\bigcap_{i=1}^{m_{UT}(n')} \mathsf{conv}(T'_i) \neq \emptyset$$

This property is preserved by replacing each point p'_i in the elements of T' by its corresponding subset Q_i, since $p'_i \in \mathsf{conv}(Q_i)$. Thus, the convex hulls of the elements of $\mathcal{T}$ have a nonempty intersection. As the set $\{Q_1, Q_2, \ldots, Q_{n'}\}$ forms a partition of P and each set Q_i corresponds to a unique point $p'_i \in P'$,

$\mathcal{T}$ is a partition of P. So far we have proved that $\mathcal{T}$ is a Tverberg partition. It remains to show the claimed tolerance. By construction of the point set P', each point $p_i' \in P'$ has depth $t+1$ with respect to its corresponding subset Q_i. By Lemma 2.11 on page 9, this is equivalent to

$$\forall p_i' \in P' : \forall R \subseteq Q_i, |R| \leq t : p_i' \in \mathsf{conv}(Q_i \setminus R)$$

Therefore, the convex hulls of the elements of $\mathcal{T}$ have a nonempty intersection even after the removal of up to t points from P.

The partitioning of P into sets $Q_1, Q_2, \ldots, Q_{\lfloor |P|/s_{CP}(t+1) \rfloor}$ in line 2 takes linear time in the size of P. The algorithm $\mathcal{A}_{CP}$ is executed on each of these sets in lines 3–4, which requires $\mathcal{O}(T_{CP}(s_{CP}(t+1)) \cdot |P|/s_{CP}(t+1))$ time in total. By our assumption, the computation of the Tverberg partition for P' takes $\mathcal{O}(T_{UT}(|P|/s_{CP}(t+1))$ time. The construction of $\mathcal{T}$ in lines 6–7 takes time $\mathcal{O}(|P|/s_{CP}(t+1))$, as each point $p_i' \in P'$ is replaced only once and each replacement can be performed in constant time if all sets are implemented as linked lists. Summing everything up, we obtain a total running time of

$$\mathcal{O}(T_{CP}(s_{CP}(t+1)) \cdot |P|/s_{CP}(t+1) + T_{UT}(|P|/s_{CP}(t+1)))$$

■

Table 4.1 on page 50 shows values for concrete approximation algorithms that are obtained by applying Theorem 4.1 to approximation algorithms for the untolerated Tverberg problem. Note that the approximation algorithms for the untolerated Tverberg problem are used for both computing points of depth $t+1$ and computing a Tverberg partition of the stabilized point set. Thus, in this analysis, $\mathcal{A}_{CP}$ and $\mathcal{A}_{UT}$ refer to the same algorithm.

An advantage of working on a stabilized point set is that each Tverberg point of the Tverberg partition $\mathcal{T}'$ for the stabilized point set is a Tverberg point of the returned Tverberg partition $\mathcal{T}$ even if we remove up to t points from P. This in an interesting property since both approximation algorithms for the untolerated Tverberg problem return a Tverberg partition together with a corresponding Tverberg point. By storing this returned Tverberg point for $\mathcal{T}'$, we obtain a Tverberg point for the returned Tverberg partition $\mathcal{T}$ that remains a Tverberg point even if up to t points are removed from P during the construction of $\mathcal{T}$ without any additional computation.

Proposition 4.2. *Let $\mathcal{A}_{UT}$ and $\mathcal{A}_{CP}$ be as in Theorem 4.1 and let $P \subset \mathbb{R}^d$ be* *a point set and $t \in \mathbb{N}$ the tolerance parameter. Let further be $\mathcal{T}$ the returned t-tolerated Tverberg partition by algorithm 4.1 on input $(P, t, \mathcal{A}_{CP}, \mathcal{A}_{UT})$ and $\mathcal{T}'$ the Tverberg partition of the stabilized point set. Then*

$$c \in \bigcap_{T_i' \in \mathcal{T}'} \mathsf{conv}(T_i') \Rightarrow \forall R \subset P, |R| \leq t : c \in \bigcap_{T_i \in \mathcal{T}} \mathsf{conv}(T_i \setminus R)$$

Proof. Let c be a Tverberg point of $\mathcal{T}'$. Fix some subset $R \subset P$ of size at most t. In the proof of Theorem 4.1, we have already argued that each stabilized point p_i' is contained in $\mathsf{conv}(Q_i \setminus R)$. By construction of $\mathcal{T}$, we thus know that $\forall T_i' \in \mathcal{T}' : T_i' \subset \mathsf{conv}(T_i \setminus R)$. Together with our assumption that c is a Tverberg point of $\mathcal{T}'$, this implies that $c \in \bigcap_{T_i \in \mathcal{T}} \mathsf{conv}(T_i \setminus R)$. ■

4.1.2 Partition Hardening

In the preceding section, we have guaranteed the tolerance of the constructed Tverberg partition by modifying points. In this section, we present two slightly different approaches, *parallel* and *sequential partition hardening*, that both modify Tverberg partitions of the input point set to guarantee tolerance. The main idea behind these approaches is to combine elements of untolerated Tverberg partitions to increase their tolerance at the expense of their size.

Parallel Partition Hardening

We partition P into $t+1$ disjoint sets of equal size and use the approximation algorithm for the untolerated Tverberg problem to obtain for each of these sets an untolerated Tverberg partition. After removing any t points of P, at least one of these partitions remains untouched. By combining each ith element of these partitions, we can guarantee a nonempty intersection of the convex hulls of the combined elements.

In lines 1–4 of Algorithm 4.2, we partition P into $t+1$ sets $Q_1, Q_2, \dots, Q_{t+1}$ of equal size and use algorithm $\mathcal{A}_{UT}$ to obtain for each set a Tverberg partition of size $m = m_{UT}(\lfloor P/(t+1) \rfloor)$. All ith elements of these partitions are combined into one element of the returned Tverberg partition in lines 5–6.

Theorem 4.3. *Let $\mathcal{A}_{UT}$ be an approximation algorithm for the untolerated Tverberg problem under the assumptions of Theorem 4.1 and $P \subset \mathbb{R}^d$ a point set. Algorithm 4.2 computes a t-tolerated Tverberg partition of P of size $m_{UT}(\lfloor P/(t+1) \rfloor)$ in time $\mathcal{O}(t \cdot T_{UT}(|P|/t))$.*

Algorithm 4.2: Parallel-Partition-Hardening

```
  input  : Point set $P \subset \mathbb{R}^d$, tolerance parameter $t$, algorithm $\mathcal{A}_{UT}$
1 Partition $P$ into sets $Q_1, Q_2, \ldots, Q_{t+1}$ of equal size;
2 $m \leftarrow m_{UT}(\lfloor |P|/(t+1) \rfloor)$;
3 foreach $i \in \{1, 2, \ldots, t+1\}$ do
4     $\mathcal{T}_i = \{T_{i,1}, T_{i,2}, \ldots, T_{i,m}\} \leftarrow \mathcal{A}_{UT}(Q_i)$;
5 foreach $j \in \{1, 2, \ldots, m\}$ do
6     $T_j \leftarrow \bigcup_{i \in \{1,2,\ldots,t+1\}} T_{i,j}$;
7 return $\{T_1, T_2, \ldots, T_m\}$;
```

Proof. We first prove that the returned set $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ is a partition of P of size $m = m_{UT}(\lfloor P/(t+1) \rfloor)$ and then show the conjectured tolerance. Since $Q_1, Q_2, \ldots, Q_{t+1}$ is a partition of P and each $\mathcal{T}_i$ is a partition of Q_i, the set $\bigcup_{i=1}^{t+1} \mathcal{T}_i$ is a partition of P. This still holds if we combine elements from $\bigcup_{i=1}^{t+1} \mathcal{T}_i$, which proves that $\mathcal{T}$ is a partition of P. As the combination does not change the size of the initial Tverberg partitions, $\mathcal{T}$ is a partition of size m.

Fix some subset $R \subseteq P$ of size at most t. Since we have partitioned P into $t+1$ sets, there is at least one set Q_i that contains no point from R. Thus, the convex hulls of the elements of $\mathcal{T}_i$ have still a nonempty intersection even if we remove R from P. By construction, each element $T_{i,j}$ is a subset of T_j. This implies, that $\cap_{j=1}^{m} \mathsf{conv}(T_j \setminus R) \neq \emptyset$. Since R was an arbitrary subset of size at most t, $\mathcal{T}$ is t-tolerated.

The initial partitioning in line 1 can be performed in time $\mathcal{O}(|P|)$. The computation of the Tverberg partitions for the initial sets in line 4 requires $\mathcal{O}(t \cdot T_{UT}(|P|/t))$ time. The combination of the elements of the Tverberg partitions in line 6 can be carried out in $\mathcal{O}(t \cdot m_{UT}(|P|/t))$ if the sets are represented as linked lists. Since the time $T_{UT}(|Q_i|)$ to construct a Tverberg partition of Q_i is at least linear in the size of the returned partition $m_{UT}(|Q_i|)$ and in the size of Q_i, this results in a total time complexity of $\mathcal{O}(t \cdot T_{UT}(|P|/t))$. ∎

We refer to this method as *parallel* partition hardening, as the computation of the Tverberg partitions for each set Q_i is completely independent of solutions for other sets Q_j $(j \neq i)$ and can be carried out in parallel.

Let us again take a look at Table 4.1 on page 50. Compared to point stabilization, we obtain approximation algorithms for the tolerated Tverberg problem that compute partitions of much greater size. Unfortunately, using this approach, we are unable to compute a point that is a Tverberg point of the constructed Tverberg partition $\mathcal{T}$ if up to t points are removed from P. However, we can still take advantage of the specific construction of $\mathcal{T}$ to compute a Tverberg point after t points are removed. In particular, we can show that Tverberg points for at least one of the initial sets $Q_1, Q_2, \ldots, Q_{t+1}$ remain Tverberg points of $\mathcal{T}$ even after up to t points are removed.

Proposition 4.4. *Let $\mathcal{A}_{UT}$ be an algorithm under the assumptions of Theorem 4.1 and $P \subset \mathbb{R}^d$ a point set. Let further be $t \in \mathbb{N}$ the tolerance parameter and $\mathcal{T}$ the returned t-tolerated Tverberg partition of algorithm 4.1 on input $(P, t, \mathcal{A}_{UT})$. Then*

$$\forall R \subset P, |R| \leq t : \exists i \in \{1, 2, \ldots, t+1\} : \bigcap_{T_{i,j} \in \mathcal{T}_i} \mathsf{conv}(T_{i,j}) \subseteq \bigcap_{T_j \in \mathcal{T}} \mathsf{conv}(T_j \setminus R)$$

where $\mathcal{T}_i$ is the Tverberg partition for Q_i that is computed by algorithm 4.1 in line 4.

Proof. Fix some subset $R \subset P$ of size at most t. Let $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{t+1}$ be the Tverberg partitions of the sets $Q_1, Q_2, \ldots, Q_{t+1}$, respectively. In the proof of Theorem 4.3, we showed that there exists at least one Tverberg partition $\mathcal{T}_i$ such that $T_{i,j} \subset T_j \setminus R$ for all $j \in \{1, 2, \ldots, m'\}$. Thus every Tverberg point of $\mathcal{T}_i$ is also a Tverberg point of $\mathcal{T}$. Since R was arbitrary, this concludes the proof. ■

This result implies a simple algorithm to determine a Tverberg point of the returned Tverberg partition after the removal of up to t points if the used approximation algorithm for the untolerated Tverberg problem returns a Tverberg point together with the computed Tverberg partition. Suppose that each point $p \in P$ has a pointer to the initial set Q_i in which it is contained. Given a subset $R \subset P$ of size at most t, we can use the pointer to the initial sets to find a Q_i from which no point is removed and return the corresponding Tverberg point.

Sequential Partition Hardening

Instead of partitioning P into $t+1$ sets and computing for each set a Tverberg partition, we can also use a Tverberg partition of the entire input set P to construct a tolerated one. Let $m = m_{UT}(|P|)$ be the size of the returned partition by $\mathcal{A}_{UT}$ on input P and $\mathcal{T}' = \{T'_1, T'_2, \ldots, T'_m\}$ be the corresponding partition. We partition $\mathcal{T}'$ into $t+1$ subsets $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{t+1}$, each of size $\lfloor m/(t+1) \rfloor$. Since $\mathcal{T}'$ is a Tverberg partition, each subset $\mathcal{T}_i$ is also a Tverberg partition of a subset of P. Thus, we can use the same method as in the parallel partition hardening reduction to construct a t-tolerated Tverberg partition for P out of $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{t+1}$. Because this time we had to compute the untolerated Tverberg partition for the entire set P in advance, we refer to this approach as *sequential* partition hardening.

Algorithm 4.3: Sequential-Partition-Hardening

```
  input : Point set $P \subset \mathbb{R}^d$, tolerance parameter $t$, algorithm $\mathcal{A}_{UT}$
1 $m' \leftarrow m_{UT}(|P|)$;
2 $\mathcal{T}' = \{T'_1, T'_2, \ldots, T'_{m'}\} \leftarrow \mathcal{A}_{UT}(Q_i)$;
3 $m \leftarrow \lfloor m'/(t+1) \rfloor$;
4 foreach $j \in \{1, 2, \ldots, m\}$ do
5     $T_j \leftarrow \bigcup_{i \in \{(m-1)(t+1)+1, (m-1)(t+1)+2, \ldots, m(t+1)\}} T'_i$;
6 return $\{T_1, T_2, \ldots, T_m\}$;
```

Theorem 4.5. *Let $\mathcal{A}_{UT}$ be an approximation algorithm for the untolerated Tverberg problem under the assumptions of Theorem 4.1 and $P \subset \mathbb{R}^d$ a point set. Algorithm 4.3 computes a t-tolerated Tverberg partition of size $\lfloor m_{UT}(|P|)/(t+1) \rfloor$ for P in time $\mathcal{O}(T_{UT}(|P|))$. Also, every Tverberg point of $\mathcal{T}'$ is a Tverberg point of $\mathcal{T}$ even if up to t points are removed from P, where $\mathcal{T}'$ is the Tverberg partition computed by algorithm 4.3 in line 2.*

Proof. The correctness of the algorithm relies on Theorem 4.3. The running time of Algorithm 4.3 is clearly dominated by the computation of the untolerated Tverberg partition in line 2.

Let c be a Tverberg point of the Tverberg partition $\mathcal{T}'$ computed by $\mathcal{A}_{UT}$ in line 2. Since each Tverberg partition $\mathcal{T}'_i$ is a subset of the elements of $\mathcal{T}'$, c is also a Tverberg point of $\mathcal{T}'_i$. In the proof of Theorem 4.3, we have argued that one of the Tverberg points of the combined partitions is still a Tverberg point of the returned partition after the removal of up to t points. In this case, all combined Tverberg partitions have a Tverberg point in common, so we know that this point is still a valid Tverberg point of the returned partition after up to t points are removed. ■

Although the sequential partition hardening reduction is very similar to the parallel version, there are subtle differences. One different aspect is the size of the obtained tolerated partition. As stated, we obtain a Tverberg partition of size $\lfloor m_{UT}(|P|)/(t+1) \rfloor$ in contrast to $m_{UT}(\lfloor |P|/(t+1) \rfloor)$. While this makes neither for the algorithm by Miller & Sheehy [7] nor for the algorithm by Mulzer

& Werner [9] a difference, as in both cases the size of the untolerated Tverberg partition is linear in the size of the input point set, in general both terms do not have to be equal.

If we compare sequential partition hardening with the parallel version in Table 4.1 on page 50, we note another difference. If the running time of the approximation algorithm for the untolerated Tverberg problem is not linear in the size of P, as it is the case for Miller & Sheehy's algorithm, the running time of the obtained approximation algorithm for the tolerated Tverberg problem is worse than the running time of the approximation algorithm obtained using parallel partition hardening.

Perhaps the main difference, which is independent of the properties of the used approximation algorithms for the untolerated Tverberg problem, is the possibility to compute a Tverberg point of the returned partition that remains a Tverberg point after the removal of up to t points. In the parallel variant, it is in general impossible to find such a point. In the sequential variant, every Tverberg point of the untolerated partition of P is also a Tverberg point of the Tverberg partitions $T'_1, T'_2, \dots, T'_{t+1}$ and thus remains a Tverberg point for the tolerated partition, even after up to t points are removed. In this sense, sequential partition hardening combines the advantage of computing a Tverberg point that remains a Tverberg point after the removal of up to t points in advance from point stabilization, with the greater size of the computed tolerated Tverberg partition from parallel partition hardening.

If we want to always compute t-tolerated Tverberg partitions that admit the computation of Tverberg points that remain Tverberg points even if up to t points are removed from P, we can show that sequential partition hardening cannot be improved in the quality of the obtained approximation algorithms.

Consider an algorithm $\mathcal{A}_{UT}$ that returns on input $P \subset \mathbb{R}^d$ a Tverberg partition of maximum size. Suppose we could use this algorithm to construct an approximation algorithm $\mathcal{A}_{TT}$ for the tolerated Tverberg problem that returns on input (P, t) a t-tolerated Tverberg partition $\mathcal{T}$ for P of size strictly greater than $m_{UT}(|P|)/(t+1)$ and there exists a Tverberg point c of $\mathcal{T}$ that remains a Tverberg point of $\mathcal{T}$ even if up to t points are removed. That is, for all subsets $R \subset P$ of size at most t, c is contained in $\bigcap_{T_i \in \mathcal{T}} \mathsf{conv}(T_i \setminus R)$. By Lemma 2.11, this implies that c has depth $t+1$ with respect to each element of $\mathcal{T}$ and thus the depth of c is strictly greater than $(t+1)m_{UT}(|P|)/(t+1) = m_{UT}(|P|)$ with respect to P. Since $\mathcal{A}_{UT}$ returns a Tverberg partition of maximum depth, $m_{UT}(|P|)$ is greater or equal to $|P|/(d+1)$. Therefore, the depth of c is strictly greater than $|P|/(d+1)$. Because P was arbitrary, this is a contradiction to the tightness of the centerpoint theorem.

4.2 Reduction to the Constant Tolerated Tverberg Problem

We refer to the problem of finding a t-tolerated Tverberg partition as constant tolerated Tverberg problem if t is a constant and not part of the input. For the special case of $t = 0$, we have developed several approximation preserving reductions in the preceding section. In this section, we develop an approximation preserving reduction for the more general case in which t can be an arbitrary constant.

In the parallel partition hardening method, we combine several Tverberg partitions for disjoint subsets to create a tolerated Tverberg partition for the entire set. This still works if applied to a set of Tverberg partitions with tolerance greater than zero. In other words, we combine Tverberg partitions for disjoint sets with some tolerance to obtain a Tverberg partition for the whole point set with higher tolerance.

Lemma 4.6. *(Combination Lemma) Let $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$ be Tverberg partitions of size m and tolerance t for disjoint point sets $P_1, P_2, \ldots, P_k \subset \mathbb{R}^d$. Then*

$$\mathcal{T} = \left\{T_i = \bigcup_{j=1}^{k} T_{j,i} \mid i \in \{1, 2, \ldots, m\}\right\}$$

is a Tverberg partition of $P = \bigcup_{j=1}^{k} P_i$ with tolerance $k(t+1)-1$ and size m, where $T_{j,i}$ is the ith element of $\mathcal{T}_j$.

Proof. Let $R \subseteq P$ be any subset of size less or equal to $k(t+1)-1$. Since we have k partitions, there exists a j such that $P_j \cap R \leq t$. By our precondition, $\mathcal{T}_j$ is t-tolerated. We thus know that $\bigcap_{i=1}^{m} \mathsf{conv}(T_{j,i} \setminus R) \neq \emptyset$. Because each element $T_{j,i}$ is contained in the corresponding element T_i of $\mathcal{T}$, the convex hulls of the elements in $\mathcal{T}$ still intersect after the removal of R. Since R was arbitrary, the claim follows. ■

Note that the parallel partition hardening method is a special case of this lemma, where the tolerance of the partitions $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$ is 0.

Lemma 2.13 by Miller & Sheehy is very similar to Lemma 4.6, as it also constructs a Tverberg partition for the entire set out of several Tverberg partitions for disjoint subsets. While our lemma preserves the size of the combined Tverberg partitions and increases the tolerance, the lemma by Miller & Sheehy increases the size but does not preserve the tolerance. Despite this difference, we can adopt the general scheme of their algorithm in conjunction with our combination lemma: We partition the input point set into small subsets of equal size and use an approximation algorithm for the constant tolerated Tverberg problem to obtain initial solutions for each subset. In a second step, we repeatedly combine these solutions to increase their tolerance until only one solution for the entire point set is left.

Algorithm 4.4 follows this approach. The parameter n' of the input defines the size of the subsets on which the approximation algorithm $\mathcal{A}_{CTT}$ for the constant tolerated Tverberg problem is applied. The parameter k defines how many of the tolerated Tverberg problems for subsets of P are combined at once using Lemma 4.6. We will later determine the optimal values for these parameters. The solutions of $\mathcal{A}_{CTT}$ for the initial sets are computed and stored in lines 1–4. The solutions array contains in index i all unused tolerated Tverberg partitions that have been computed after i combination steps. By unused we mean that those partitions have not been used yet during the combination steps. Note that all partitions in solutions$[i]$ have the same tolerance. The combination itself is carried out in lines 7–10 and is repeated until there are less than k partial solutions left of the same tolerance. The outer repeat-until-loop (line 6) terminates if no combination was performed. That is, when each cell of the solutions array contains strictly less than k elements. Finally, all partial solutions are combined using the parallel partition hardening method to construct one partition for the entire input set in line 13.

Theorem 4.7. *Given a point set $P \subseteq \mathbb{R}^d$, let $\mathcal{A}_{CTT}$ be an algorithm that returns a t_{CTT}-tolerated partition of size $m_{CTT}(|P|)$, where t_{CTT} is a constant. Then Algorithm 4.4 returns a $\left(\frac{|P|+n'}{kn'}(t_{CTT}+1)-1\right)$-tolerated Tverberg partition of P of size $m_{CTT}(n')$ in time $\mathcal{O}\left(T_{CTT}(n') \cdot |P|/n' + k \cdot m_{CTT}(n') \log_k\left(|P|/n'\right)\right)$, where $T_{CTT}(n')$ is the running time of algorithm $\mathcal{A}_{CTT}$ if applied to a set of size of at most n'.*

Proof. We first show that algorithm 4.4 always returns a Tverberg partition for P of size $m_{CTT}(n')$ and prove afterwards the tolerance and the running time of the algorithm.

We prove that the output is a Tverberg partition of P by showing the following invariant: The solutions array always contains Tverberg partitions of size $m_{CTT}(n')$ of disjoint subsets that together form a partition of P. This holds in the beginning of the algorithm (lines 1–4), as the solutions array contains the results of $\mathcal{A}_{CTT}$ applied to each set of a partition of P. During each combination step (lines 7–10), the combined Tverberg partitions $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$

Algorithm 4.4: Constant-Tolerated-Tverberg-Reduction

```
input  : Point set $P$, approximation algorithm for the constant tolerated
         Tverberg problem $\mathcal{A}_{CTT}$, base set size $n'$, branching factor $k$
1  $\{P_1, P_2, \dots, P_{\lfloor |P|/n' \rfloor}\} \leftarrow$ partition of $P$ into sets of size $n'$;
2  solutions $\leftarrow$ array of sets of tolerated Tverberg partitions for subsets of $P$;
3  foreach $i \in \{1, 2, \dots, \lfloor |P|/n' \rfloor\}$ do
4      solutions[0] $\leftarrow$ solutions[0] $\cup \{\mathcal{A}_{CTT}(P_i)\}$;
5  level $\leftarrow 0$;
6  repeat
7      while |solutions[level]| $\geq k$ do
8          $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k \leftarrow$ remove any $k$ elements from solutions[level];
9          $\mathcal{T} \leftarrow$ combination of $\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k$ using Lemma 4.6;
10         solutions[level + 1] $\leftarrow$ solutions[level + 1] $\cup \{\mathcal{T}\}$;
11     level $\leftarrow$ level + 1;
12 until no combination was performed;
13 $\mathcal{T} \leftarrow$ combination of all tolerated Tverberg partitions in solutions using the
   parallel partition hardening method;
14 return $\mathcal{T}$;
```

are removed in line 8 and will not be part of the combination anymore. The obtained partition $\mathcal{T}$ of this combination is added to the solutions array. Let P'_i be the subset of P for which $\mathcal{T}_i$ is a Tverberg partition. By the correctness of Lemma 4.6, $\mathcal{T}$ is a Tverberg partition of $\bigcup_{i=1}^{k} P'_i$ and is of the same size as the combined partitions. Thus, at the end of the combination, the invariant holds.

This implies, that the final combination step in line 13 indeed returns a Tverberg partition of P of size $m_{CTT}(n')$.

We continue with proving the tolerance of the final Tverberg partition. Each Tverberg partition in solutions$[i]$ is the result of i combination steps. If t' is the tolerance of the combined partitions, the resulting partition has tolerance $k(t'+1)-1$. Due to algorithm $\mathcal{A}_{CTT}$, the initial Tverberg partitions have tolerance $t_{\mathcal{A}_{CTT}}$. This leads to the following recurrence relation

$$\begin{aligned}\text{tolerance}(i) &= k \cdot (\text{tolerance}(i-1)+1) - 1\\ \text{tolerance}(0) &= t_{\mathcal{A}_{CTT}}\end{aligned}$$

which resolves to

$$\begin{aligned}
\mathsf{tolerance}(i) &= k\Big(k(\mathsf{tolerance}(i-2)+1)-1+1\Big)-1 && \Leftrightarrow\\
\mathsf{tolerance}(i) &= k^2\left(\mathsf{tolerance}(i-2)+1\right)-1 && \Leftrightarrow\\
\mathsf{tolerance}(i) &= k^i(t_{\mathcal{A}_{CTT}}+1)-1
\end{aligned}$$

To give a lower bound on the tolerance of the returned partition, we need to know the maximum index of the $\mathsf{solutions}$ array. Let us analyze the size of a subset of P that corresponds to a partition in $\mathsf{solutions}[i]$. The initial point sets are of size n'. Let again be $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$ the combined partitions, where P_i is the corresponding point set to $\mathcal{T}_i$. The resulting partition $\mathcal{T}$ then corresponds to the point set $\bigcup_{j=1}^{k} P_i$. Thus, the number of points of each Tverberg partition in $\mathsf{solutions}[i]$ is $k^i n'$. The algorithm advances in line 11 only if there are strictly less than k elements left in $\mathsf{solutions}[\mathsf{level}]$. In the end, each cell of the $\mathsf{solutions}$ array contains strictly less than k partitions. We have already proved, that the corresponding subsets to each partition in the $\mathsf{solutions}$ array form a partition of P. This leads to the following upper bound on the number of points in P

$$|P| \le \sum_{i=0}^{h}(k-1)(k^i \cdot n')$$

where h is the maximum index of the $\mathsf{solutions}$ array at the termination. We can use this inequality to lower bound h.

$$\begin{aligned}
|P| &\le (k-1)n'\sum_{i=0}^{h} k^i && \Leftrightarrow\\
&= (k-1)n'\frac{k^{h+1}-1}{k-1} && \Leftrightarrow\\
\log_k\left(\frac{|P|}{n'}+1\right)-1 &\le h
\end{aligned}$$

This implies that $\mathsf{solutions}[\log_k\left(\frac{|P|}{n'}+1\right)-1]$ contains at least one Tverberg partition. Since this partition is used to build the returned partition $\mathcal{T}$, $\mathcal{T}$ has to be of tolerance at least $k^{\log_k\left(\frac{|P|}{n'}+1\right)-1}(t_{CTT}+1)-1 = \frac{|P|+n'}{kn'}(t_{CTT}+1)-1$.

The only remaining point of the proof is the running time of algorithm 4.4. Computing the initial partitions takes $\mathcal{O}(T_{CTT}(n')\cdot|P|/n')$ time. The crucial part are both loops. The inner while loop removes and combines in each step k partitions from solutions[level]. Both the removal and the combination can be carried out within $\mathcal{O}(k \cdot m_{CTT}(n'))$ time if the sets are represented as linked lists. The outer loop increases the level in each iteration. We need to upper bound the maximum index h of the solutions array. There can be no partitions whose corresponding point set is greater than P. Using our result about the number of points of the partitions in solutions$[h]$, we get the following inequality

$$|P| \geq k^{h-1}n' \Leftrightarrow h \leq \log_k \left(\frac{|P|}{n'}\right) + 1$$

This leads to a complexity of $\mathcal{O}\left(k \cdot m_{CTT}(n') \log_k \left(|P|/n'\right)\right)$ for both loops. The running time of the combination in line 13 can be upper bounded by $\mathcal{O}(|P|)$, which results in the conjectured total time complexity of

$$\mathcal{O}\left(T_{CTT}(n') \cdot |P|/n' + k \cdot m_{CTT}(n') \log_k \left(|P|/n'\right)\right)$$

■

Equipped with this result, we are now ready to determine the optimal values for the branching factor k and the size n' of the initial point sets. As with our previous reductions, we always want Algorithm 4.4 to return a Tverberg partition of the desired tolerance t. We also want to choose n' maximal without violating the tolerance constraint so that algorithm $\mathcal{A}_{CTT}$ returns a tolerated Tverberg partition of maximal size for the initial point sets .

Proposition 4.8. *Under the preconditions of Theorem 4.7, Algorithm 4.4 returns a t-tolerated partition of size $m_{CTT}(|P|(t_{CTT}+1)/(2t-t_{CTT}+1))$ if $n' = |P|(t_{CTT}+1)/(2t-t_{CTT}+1)$ and $k = 2$. In general, there is no assignment to n' and k for which Algorithm 4.4 returns a t-tolerated Tverberg partition of greater size.*

Proof. The constraint on the tolerance of the returned partition can be used to upper bound the size of the initial point sets.

$$t \leq \frac{|P| + n'}{kn'}(t_{CTT}+1) - 1 \qquad \Leftrightarrow$$

$$(t+1)\, n' \leq \frac{|P| + n'}{k}(t_{CTT}+1) \qquad \Leftrightarrow$$

$$\frac{k(t+1)-(t_{CTT}+1)}{k}n' \leq \frac{|P|(t_{CTT}+1)}{k} \quad\quad \Leftrightarrow$$

$$n' \leq \frac{|P|(t_{CTT}+1)}{k(t+1)-(t_{CTT}+1)}$$

As m_{CTT} is a monotonically increasing function, the size of the returned partition is maximized if k is chosen as small as possible. We can safely assume, that $t > t_{CTT}$ since otherwise we could use the approximation algorithm for the constant tolerated Tverberg problem directly. Choosing $k = 2$ results in $n' = |P|(t_{CTT}+1)/(2t - t_{CTT}+1)$ as claimed. ■

One important step in both approximation algorithms for the untolerated Tverberg problem is to identify points that are not needed for the convex hulls of the elements of the untolerated Tverberg partition in order to have a nonempty intersection. In section 2.3, we refered to this step as *pruning*. Unnecessary points can be removed and fed back into the algorithm, which virtually increases the size of the initial point set and thus the size of the returned Tverberg partition. Unfortunately, in a t-tolerated Tverberg partition it is unclear how points that are not needed for the convex hulls to have a nonempty intersection *even if we remove t points from P* can be identified. Let us take again a look at Carathéodory's theorem, which is used by Miller & Sheehy and Mulzer & Werner to prune points. It limits the size of the elements of the untolerated Tverberg partition to less than $d+1$. This can obviously not work on tolerated Tverberg partitions, as the tolerance t could be greater than $d+1$. However, Montejano and Oliveros [8] proved a tolerated version of Carathéodory's theorem:

Theorem 4.9 (Tolerance Carathéodory Theorem [8, Theorem 4.1])**.** *Let $P \subset \mathbb{R}^d$ be a set, $c \in \mathbb{R}^d$ a point, and $t \in \mathbb{N}$ the tolerance parameter. Then c is contained in $\bigcap_{R\subseteq P, |R|\leq t} \mathsf{conv}(P \setminus R)$ if and only if there is some subset $P' \subseteq P$ of size of at most $\eta(d+1, t+1)$ such that $c \in \bigcap_{R\subseteq P, |R|\leq t} \mathsf{conv}(P' \setminus R)$, where η is the Erdős-Gallai bound.*

Unfortunately, this result is of no help in pruning tolerated Tverberg partitions. Let $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ be a t-tolerated Tverberg partition. We want to use Theorem 4.9 to upper bound the size of each element in $\mathcal{T}$. The main problem is, that in general there is no point c that satisfies the conditions of Theorem 4.9 as there is in general no point of depth $t+1$ with respect to *each* element of $\mathcal{T}$. Otherwise it would lead to an improvement of the centerpoint bound, which is a contradiction to the tightness of the bound. But even if such a point existed, the theorem is too weak to improve the current reduction. Tuza [14] proved a sharp upper bound on the Erdős-Gallei bound which is exponential in the first parameter. This is in our case the dimension. However, this is worse than our upper bound on the total number of points of the constructed tolerated Tverberg partition in the proof of Theorem 4.7.

Algorithm	**Size**	**Running Time**
Theorem 4.1 with Miller-Sheehy	$\lvert P\rvert/4(t+1)(d+2)^4$	$\mathcal{O}(\lvert P\rvert(td^2)^{c_{ms}\log(d)-1} + (\lvert P\rvert/td^2)^{c_{ms}\log(d)})$
Theorem 4.1 with Mulzer-Werner	$\lvert P\rvert/16(t+1)(d+1)^6$	$\mathcal{O}(d^{c_{mw}\log(d)}\,\lvert P\rvert)$
Theorem 4.3 with Miller-Sheehy	$\lvert P\rvert/2(t+2)(d+1)^2$	$\mathcal{O}(t\,(\lvert P\rvert/t)^{c_{ms}\log(d)})$
Theorem 4.3 with Mulzer-Werner	$\lvert P\rvert/4(t+1)(d+1)^3$	$\mathcal{O}(d^{c_{mw}\log(d)}\lvert P\rvert)$
Theorem 4.5 with Miller-Sheehy	$\lvert P\rvert/2(t+1)(d+1)^2$	$\mathcal{O}(\lvert P\rvert^{c_{ms}\log(d)})$
Theorem 4.5 with Mulzer-Werner	$\lvert P\rvert/4(t+1)(d+1)^3$	$\mathcal{O}(d^{c_{mw}\log(d)}\lvert P\rvert)$
Theorem 4.7 with Miller-Sheehy	$\lvert P\rvert/(4t+2)(d+1)^2$	$\mathcal{O}(t\,(\lvert P\rvert/t)^{c_{ms}\log(d)} + \lvert P\rvert\log(t)/td^2)$
Theorem 4.7 with Mulzer-Werner	$\lvert P\rvert/(8t+4)(d+1)^3$	$\mathcal{O}(d^{c_{mw}\log(d)}\lvert P\rvert)$

Table 4.1: Quality of approximation algorithms for the tolerated Tverberg problem that are obtained by reduction to concrete approximation algorithms for the untolerated Tverberg problem. The columns “Size” and “Running Time” refer to the computation of a t-tolerated Tverberg partition for a point set $P \subset \mathbb{R}^d$, where c_{ms} and c_{mw} are constants specific to the algorithms by Miller & Sheehy [7] and Mulzer & Werner [9], respectively.

5 Testing-Tolerated-Tverberg

Teng proved in his Ph.D. thesis [13] that testing whether a point is a centerpoint is coNP-complete if the dimension is part of the input. He also showed that the problem to test whether there exists a Tverberg partition containing a given point is NP-complete if the dimension is part of the input. We refer to these problems as Testing-Center and Testing-Tverberg.

Naturally, the question arises how hard it is to decide whether a given Tverberg partition is t-tolerated, where the parameter t and the dimension is part of the input. We call this decision problem Testing-Tolerated-Tverberg. Note that this is not a generalization of Testing-Tverberg as the naming suggests. There is no obvious polynomial-time reduction of Testing-Tolerated-Tverberg to Testing-Tverberg, as the former focuses on a specific partition and the latter on the existence of any partition. However, we can use the Testing-Center problem to prove coNP-completeness of Testing-Tolerated-Tverberg.

Proposition 5.1. *Testing-Tolerated-Tverberg is coNP-complete if the dimension d, the size m of the Tverberg partition and the conjectured tolerance t are part of the input.*

Proof. We first check that Testing-Tolerated-Tverberg is indeed contained in the complexity class coNP. Let $\mathcal{T}$ be a Tverberg partition of $P \subset \mathbb{R}^d$ that is conjectured to have tolerance t. A witness to $\mathcal{T}$ not being a t-tolerated Tverberg partition would be any subset $R \subseteq P$ of size at most t, such that $\bigcap_{T_i \in \mathcal{T}} \mathsf{conv}(T_i \setminus R) = \emptyset$. Consider the linear program $\mathcal{L}$ defined by the following constraints for each element T_i of $\mathcal{T}$, where $p_{i,j}$ denotes the jth point in $T_i \setminus R$:

$$\begin{aligned} \alpha_{i,1}\, p_{i,1} + \alpha_{i,2}\, p_{i,2} + \ldots + \alpha_{i,|T_i \setminus R|}\, p_{i,|T_i \setminus R|} - x &= 0 \\ \alpha_{i,1} + \alpha_{i,2} + \ldots + \alpha_{i,|T_i \setminus R|} &= 1 \\ \forall j \in \{1, 2, \ldots, |T_i \setminus R|\} : \alpha_{i,j} &\geq 0 \end{aligned}$$

$\mathcal{L}$ has a feasible solution if and only if $\bigcap_{T_i \in \mathcal{T}} \mathsf{conv}(T_i \setminus R) = \emptyset$. Therefore, we can check if R is a witness by showing that the constrains in $\mathcal{L}$ are inconsistent. Since $\mathcal{L}$ is defined by $m(d+1) + |P| - |R|$ constraints in $|P| - |R| + 1$ variables and linear programs can be solved in polynomial time in the number of the constraints and variables, we can check if R is a witness in polynomial time the length of $\mathcal{T}$. Therefore, Testing-Tolerated-Tverberg $\in$ coNP.

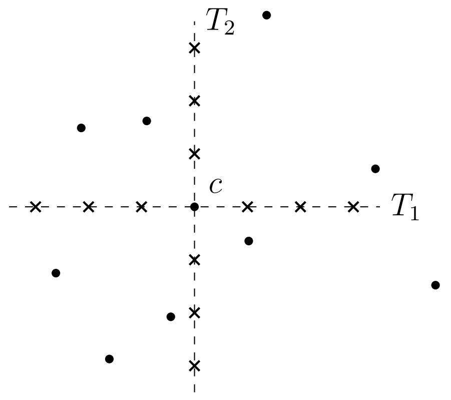

Figure 5.1: Reduction of Testing-Tolerated-Tverberg to Testing-Center

Let (P, c) be an input to Testing-Center. We construct a partition that is a Tverberg partition with tolerance $t = \lceil |P|/(d+1) \rceil - 1$ if and only if c is a centerpoint. Consider the following two sets

$$\begin{aligned} T_1 =& \{c + i \cdot (1, 0 \dots 0)^T \mid i \in \{-(t+1), -t, \dots, -1, 1, 2, \dots, t+1\}\} \\ T_2 =& \{c + i \cdot (0, 1, 0 \dots 0)^T \mid i \in \{-(t+1), -t, \dots, -1, 1, 2, \dots, t+1\}\} \end{aligned}$$

Both sets contain points on a line through c, having $t + 1$ points on each side of c. Figure 5.1 presents an example in two dimensions. The dots are points from the original point set P, crosses mark the generated points in T_1 and T_2. We claim $\mathcal{T} = \{P, T_1, T_2\}$ to be the desired partition.

Assume c is a centerpoint of P. By definition, c has depth at least $|P|/(d+1)$. Equivalently, c is contained in the convex hull of P after the removal of any $\lceil |P|/(d+1) \rceil - 1 = t$ points from P. Since the sets T_1 and T_2 contain $t + 1$ points on both sides of a line through c, c is contained in the convex hulls $\mathsf{conv}(T_1)$ and $\mathsf{conv}(T_2)$, even if up to t points from $T_1 \cup T_2$ are removed. We therefore know that

$$\forall R \subseteq (P \cup T_1 \cup T_2), |R| \leq t : c \in \mathsf{conv}(P \setminus R) \cap \mathsf{conv}(T_1 \setminus R) \cap \mathsf{conv}(T_2 \setminus R)$$

Thus, $\mathcal{T}$ forms a Tverberg partition with tolerance t for $P \cup T_1 \cup T_2$.

We now prove the opposite direction. Let $\mathcal{T}$ be constructed as above for the input (P, c) to the Testing-Center problem. Assume $\mathcal{T}$ is a t-tolerated Tverberg partition. We know that $c = \mathsf{conv}(T_1) \cap \mathsf{conv}(T_2)$ by construction of T_1, T_2. Together with the definition of tolerated Tverberg partitions, we obtain

$$\begin{aligned}\forall R \subseteq (P \cup T_1 \cup T_2), |R| \leq t : \mathsf{conv}(P \setminus R) \cap \mathsf{conv}(T_1 \setminus R) \cap \mathsf{conv}(T_2 \setminus R) = c \\ \Rightarrow \forall R \subseteq P, |R| \leq t : c \in \mathsf{conv}(P \setminus R)\end{aligned}$$

This is equivalent to c having depth at least $t+1 = (\lceil |P|/(d+1) \rceil - 1) + 1 = \lceil |P|/(d+1) \rceil$ with respect to P, or c being a centerpoint of P.

We have proved the correctness of the reduction. The size of T_1 and T_2 depends upon the parameter t. However, we can safely assume t to be less than $|P|$ since there is no t-tolerated Tverberg partition of P if $t \geq |P|$. Thus both generated sets T_1 and T_2 can be constructed in polynomial time in the length of the input, which concludes the proof. ■

In the proof of Proposition 5.1, we have constructed a Tverberg partition of size 3 that is t-tolerated if and only if c is a centerpoint of P, where $t = \lceil |P|/(d+1) \rceil - 1$. We can prove that even testing whether a given Radon partition is of some tolerance is coNP-complete.

In the proof of Proposition 5.1, both constructed sets T_1 and T_2 are necessary for the reduction. If we drop one of these sets, for example T_2, then in general $\{c\}$ is a proper subset of $\mathsf{conv}(P) \cap \mathsf{conv}(T_1)$. Thus even if the Tverberg partition $\{P, T_1\}$ is of some tolerance t, Proposition 2.12 implies that c does not have to be contained in the intersection of $\mathsf{conv}(P \setminus R) \cap \mathsf{conv}(T_1 \setminus R)$ for every subset $R \subset P \cup T_1$ of size at most t. In other words, we are not able to relate the tolerance of the partition $\{P, T_1\}$ with the depth of c with respect to P. This problem is not specific to the choice of T_1, as there is no set $T_* \subset \mathbb{R}^d, |T| > 1$ such that $|\mathsf{conv}(P) \cap \mathsf{conv}(T_*)| = 1$ if $\mathsf{dim}(\mathsf{aff}(P)) = d$, where $\mathsf{aff}(P)$ denotes the affine subspace spanned by P. However, if we introduce a new dimension, it is easy to find such a set.

Theorem 5.2. *Testing-Tolerated-Radon is coNP-complete if the dimension d and the conjectured tolerance t are part of the input.*

Proof. We prove again the hardness by a reduction to Testing-Center. Let $P \subset \mathbb{R}^d, c \in \mathbb{R}^d$ be an input to Testing-Center. We embed the vector space $\mathbb{R}^d$ in $\mathbb{R}^{d+1}$ by identifying it with the hyperplane $h_0 \subset \mathbb{R}^{d+1}$ that contains all points in $\mathbb{R}^{d+1}$ whose last coordinate is 0.

Similar to the construction of T_1, T_2 in the proof of Proposition 5.1, we can use the new dimension to construct a set T whose convex hulls intersect the convex hull of P only in c. Let l be the line that is orthogonal to h_0 and passes through c. Let T^- and T^+ be two sets that consist of any $t+1$ points in $l \cap h_0^-$ and $l \cap h_0^+$, respectively. We claim that $\{P, T\}$ is a Radon partition for $P \cup T$ with tolerance $t = \lceil |P|/d+1 \rceil - 1$ if and only if c is a centerpoint of P, where $T = T^- \cup T^+$.

"$\Rightarrow$": Assume $\{P, T\}$ is a t-tolerated Radon partition. By construction of T, the intersection of $\mathsf{conv}(P)$ and $\mathsf{conv}(T)$ consists only of c. Thus, c is contained in the intersection of both convex hulls even if any subset of size at most t is removed.

$$\forall R \subset P' \cup T, |R| \leq t : c' = \mathsf{conv}(P' \setminus R) \cap \mathsf{conv}(T \setminus R)$$

Lemma 2.11 then implies that c has depth $t+1$ with respect to P. Since $t + 1 = \lceil P/d + 1 \rceil$, c is a centerpoint for P.

"$\Leftarrow$": Assume c is a centerpoint for P. By definition, each closed half-space contains at least $\lceil |P|/d + 1 \rceil = t + 1$ points of P. Thus, c is contained in the convex hull of P even if any t points from P are removed. Since T contains $t + 1$ points on both sides of a line through c, c is also contained in $\mathsf{conv}(T)$ if any t points from T are removed. Together, this implies that $\{P, T\}$ is a t-tolerated Radon partition for $P \cup T$. ■

6 Conclusions

Due to the lack of algorithms for the tolerated Tverberg problem, the intention of this thesis was to develop algorithms to compute tolerated Tverberg partitions. By generalizing a simple algorithm for the tolerated Radon problem in one dimension, we managed to construct an algorithm which solves the tolerated Tverberg problem in one dimension. This algorithm also improved the Soberón-Strausz bound in one dimension and the new bound could be proven to be tight. Using a dimension reduction, the algorithm can be applied to multi-dimensional data. This has again led to an improvement of the Soberón-Strausz bound in two dimension if m and t are of sufficient size. However, the one-dimensional algorithm together with the dimension reduction is not well-suited for high-dimensional data.

To approach the problem in higher dimensions, we have developed two kinds of approximation preserving reductions.

First, there are three reduction to the untolerated Tverberg problem that led, based on the existing approximation algorithms for the untolerated Tverberg problem, to approximation algorithms for the tolerated Tverberg problem whose running time is not exponential in the dimension. The third one, sequential partition hardening, combines the benign properties of algorithms obtained by the former presented approximation preserving reductions: As with point stabilization, the returned t-tolerated Tverberg partitions admit the computation of a Tverberg point that remains a Tverberg point even after the removal of up to t points, while at the same time the greater size of the partition achieved by parallel partition hardening is retained.

The second kind, the approximation preserving reduction to the constant tolerated problem, is based on Miller & Sheehy's algorithm, but with a replaced combination lemma. Compared with the other reductions, it leads to better results if it is applied to algorithms that compute Tverberg partitions that already have some guaranteed tolerance greater than 0.

With the aid of the approximation preserving reductions, we can thus benefit from the existing algorithms for the untolerated Tverberg problem and any future progress.

Concerning the hardness of the tolerated Tverberg problem, the complexity of computing a tolerated Tverberg partition is still unknown. However, we could prove that a related decision problem, testing whether a given Tverberg partition is of some tolerance, is coNP-complete if the dimension is part of the input by a reduction to Testing-Center.

Another open point is the tightness of the Soberón-Strausz bound. The improvement in one dimension and the improvement in a large range of values m, t in two dimensions are indications that the bound is in general not tight.

At last, a pruning argument for tolerated Tverberg partitions is still missing. Such an argument is important for both approximation algorithms for the untolerated Tverberg problem, as it can be used to improve the size of the constructed Tverberg partitions. It would be desirable to find a similar argument for tolerated Tverberg partitions to further improve the developed approximation algorithms.

References

[1] Pankaj K. Agarwal, Micha Sharir, and Emo Welzl. Algorithms for center and Tverberg points. *ACM Transactions on Algorithms*, 5(1):1–20, 2008.

[2] Timothy M. Chan. An optimal randomized algorithm for maximum Tukey depth. In *Proceedings of the 15th annual ACM-SIAM symposium on Discrete Algorithms*, pages 430–436. Society for Industrial and Applied Mathematics, 2004.

[3] Kenneth L. Clarkson, David Eppstein, Gary L. Miller, Carl Sturtivant, and Shang-Hua Teng. Approximating center points with iterated Radon points. In *Proceedings of the 9. annual symposium on Computational Geometry*, pages 91–98. ACM, 1993.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* The MIT Press, 3rd edition, 2009.

[5] Natalia García Colín. *Applying Tverberg Type Theorems to Geometric Problems.* PhD thesis, University College London, 2007.

[6] Jiri Matoušek. *Lectures on Discrete Geometry.* Springer, 1st edition, 2002.

[7] Gary L. Miller and Donald R. Sheehy. Approximate centerpoints with proofs. *Computational Geometry*, 43:647–654, 2010.

[8] Luis Montejano and Deborah Oliveros. Tolerance in Helly-type theorems. *Discrete & Computational Geometry*, 45:348–357, 2011.

[9] Wolfgang Mulzer and Daniel Werner. Approximating Tverberg points in linear time for any fixed dimension. In *Proceedings of the 28th annual symposium on Computational Geometry*, pages 303–310. ACM, 2012.

[10] Richard Rado. A theorem on general measure. *Journal of the London Mathematical Society*, 1:291–300, 1946.

[11] Karanbir Sarkaria. Tverberg's theorem via number fields. *Israel Journal of Mathematics*, 79:317–320, 1992.

[12] Pablo Soberón and Ricardo Strausz. A generalisation of Tverberg's theorem. *Discrete & Computational Geometry*, 47:455–460, 2012.

[13] Shang-Hua Teng. *Points, spheres, and separators: a unified geometric approach to graph partitioning.* PhD thesis, Carnegie Mellon University Pittsburgh, 1992.

[14] Zsolt Tuza. Minimum number of elements representing a set system of given rank. *Journal of Combinatorial Theory, Series A*, 52(1):84–89, 1989.

[15] Helge Tverberg. A generalization of Radon's theorem. *Journal of the London Mathematical Society*, 41:123–128, 1966.

[16] Helge Tverberg. A generalization of Radon's theorem II. *Bulletin of the Australian Mathematical Society*, 24:321–325, 1981.

[17] Helge Tverberg and Siniša Vrecica. On generalizations of Radon's theorem and the ham sandwich theorem. *European Journal of Combinatorics*, 14:259–264, 1993.

[18] Daniel Werner. *Computational Aspects of some Problems from Discrete Geometry in Higher Dimensions.* PhD thesis, Freie Universität Berlin, 2013.

[19] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, 1st edition, 2011.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben. Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 19.03.2013

Yannik Stein