Discrete Comput Geom (2013) 50:520–535
DOI 10.1007/s00454-013-9528-7

Approximating Tverberg Points in Linear Time for Any Fixed Dimension

Wolfgang Mulzer · Daniel Werner

Received: 15 November 2012 / Revised: 4 June 2013 / Accepted: 25 June 2013 /
Published online: 25 July 2013

Abstract Let $P \subseteq \mathbb{R}^d$ be a d-dimensional n-point set. A *Tverberg partition* is a partition of P into r sets $P_1, \ldots, P_r$ such that the convex hulls $\operatorname{conv}(P_1), \ldots, \operatorname{conv}(P_r)$ have non-empty intersection. A point in $\bigcap_{i=1}^{r} \operatorname{conv}(P_i)$ is called a *Tverberg point* of depth r for P. A classic result by Tverberg shows that there always exists a Tverberg partition of size $\lceil n/(d+1) \rceil$, but it is not known how to find such a partition in polynomial time. Therefore, approximate solutions are of interest. We describe a deterministic algorithm that finds a Tverberg partition of size $\lceil n/4(d+1)^3 \rceil$ in time $d^{O(\log d)} n$. This means that for every fixed dimension we can compute an approximate Tverberg point (and hence also an approximate *centerpoint*) in *linear* time. Our algorithm is obtained by combining a novel lifting approach with a recent result by Miller and Sheehy (Comput Geom Theory Appl 43(8):647–654, 2010).

Keywords Discrete geometry · Tverberg theorem · Centerpoint · Approximation · High dimension

1 Introduction

In many applications (such as statistical analysis or finding sparse geometric separators in meshes) we would like to have a way to generalize the one-dimensional notion of

A preliminary version appeared as W. Mulzer and D. Werner, *Approximating Tverberg Points in Linear Time for Any Fixed Dimension* in Proceedings of 28th SoCG, pp. 303–310, 2012.

W. Mulzer (✉) · D. Werner
Institut für Informatik, Freie Universität Berlin, Berlin, Germany
e-mail: mulzer@inf.fu-berlin.de
http://page.mi.fu-berlin.de/mulzer

D. Werner
e-mail: dwerner@mi.fu-berlin.de
http://page.mi.fu-berlin.de/dawerner

a median to higher dimensions. A natural way to do this uses the notion of *halfspace depth* (or *Tukey depth*).

Definition 1.1 Let P be a finite set of points in $\mathbb{R}^d$, and let $c \in \mathbb{R}^d$ be a point (not necessarily in P). The *halfspace depth of c with respect to P* is

$$\min_{\text{halfspace } h,\, c \in h} |h \cap P|.$$

The *halfspace depth of P* is the maximum halfspace depth that any point $c \in \mathbb{R}^d$ can achieve.

A classic result in discrete geometry, the centerpoint theorem, claims that for every d-dimensional point set P with n points, there exists a *centerpoint*, i.e., a point $c \in \mathbb{R}^d$ with halfspace depth at least $n/(d+1)$ [6,15]. There are point sets where this bound cannot be improved.

However, if we actually want to compute a centerpoint for a given point set efficiently, the situation becomes more involved. For $d = 2$, a centerpoint can be found deterministically in linear time [9]. For general d, we can compute a centerpoint in $O(n^d)$ time using linear programming, since Helly's theorem implies that the set of all centerpoints can be described as the intersection of $O(n^d)$ halfspaces [7]. Chan [2] shows how to improve this running time to $O(n^{d-1})$ with the help of randomization. He actually solves the apparently harder problem of finding a point with maximum halfspace depth. If the dimension is not fixed, a result by Teng shows that it is coNP-hard to check whether a given point is a centerpoint [17].

However, as d grows, a running time of $n^{\Omega(d)}$ is not feasible. Hence, it makes sense to look for faster approximate solutions. A classic approach uses ε-approximations [3]: in order to obtain a point of halfspace depth $n(1/(d+1) - \varepsilon)$, take a random sample $A \subseteq P$ of size $O((d/\varepsilon^2)\log(d/\varepsilon))$ and compute a centerpoint for A via linear-programming. This gives the desired approximation with constant probability, and the running time after the sampling step is constant for fixed d. What more could we possibly wish for? For one, the algorithm is Monte-Carlo: with a certain probability, the reported point fails to be a centerpoint, and we know of no fast algorithm to check its validity. This problem can be solved by constructing the ε-approximation deterministically [3], at the expense of a more complicated algorithm. Nonetheless, in either case the resulting running time grows exponentially with d, an undesirable feature for large dimensions.

This situation motivated Clarkson et al. [4] to look for more efficient randomized algorithms for approximate centerpoints. They give a simple probabilistic algorithm that computes a point of halfspace depth $\Omega(n/(d+1)^2)$ in time $O(d^2(d\log n + \log(1/\delta))^{\log(d+2)})$, where δ is the error probability. They also describe a more sophisticated algorithm that finds such a point in time polynomial in n, d, and $\log(1/\delta)$. Both algorithms are based on a repeated algorithmic application of Radon's theorem (see below). Unfortunately, there remains a probability of δ that the result is not correct, and we do not know how to detect failure efficiently.

Thus, more than ten years later, Miller and Sheehy [13] launched a new attack on the problem. Their goal was to develop a deterministic algorithm for approximating

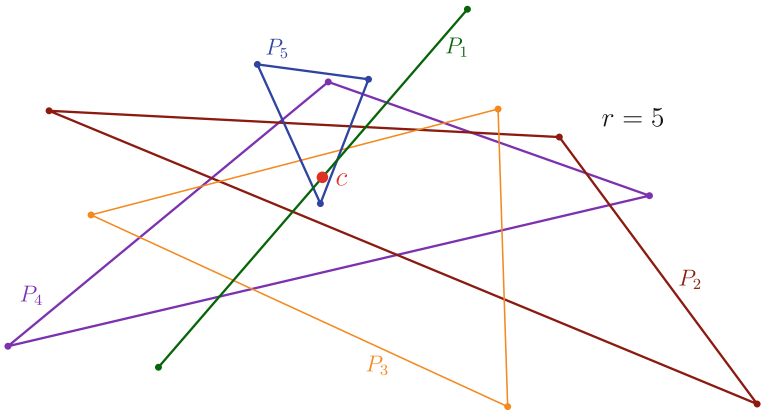

Fig. 1 The point c has Tverberg depth $r = 5$ (Color figure online)

centerpoints whose running time is subexponential in the dimension. For this, they use a different proof of the centerpoint theorem that is based on a result by Tverberg: any d-dimensional n-point set can be partitioned into $r = \lceil n/(d+1) \rceil$ sets $P_1, \ldots, P_r$ such that the convex hulls $\mathrm{conv}(P_1), \ldots, \mathrm{conv}(P_r)$ have nonempty intersection. Such a partition is called a *Tverberg partition* of P. By convexity, any point in $\bigcap_{i=1}^{r} \mathrm{conv}(P_i)$ must be a centerpoint.

More generally, we say that a point $c \in \mathbb{R}^d$ has *Tverberg depth* r' with respect to P if there is a partition of P into r' sets such that c lies in the convex hull of each set. We also call c an *approximate Tverberg point* (of depth r'); see Fig. 1.

Miller and Sheehy describe how to find $\lceil n/2(d+1)^2 \rceil$ disjoint subsets of P and a point $c \in \mathbb{R}^d$ such that each subset contains $d+1$ points and has c in its convex hull. Hence, c constitutes an approximate Tverberg point for P (and thus also an approximate centerpoint), and the subsets provide a certificate for this fact. The algorithm is deterministic and runs in time $n^{O(\log d)}$. At the same time, it is the first algorithm that also finds an approximate Tverberg partition of P. The running time is subexponential in d, but it is still the case that n is raised to a power that depends on d, so the parameters n and d are not separated in the running time.

In this paper, we show that the running time for finding approximate Tverberg partitions (and hence approximate centerpoints) can be improved. In particular, we show how to find a Tverberg partition with $\lceil n/4(d+1)^3 \rceil$ sets in deterministic time $d^{O(\log d)}n$. This is linear in n for any fixed dimension, and the dependence on d is only quasipolynomial.

1.1 Some Discrete Geometry

We begin by recalling some basic facts and definitions from discrete geometry [11]. A classic fact about convexity is Radon's theorem.

Theorem 1.2 (Radon's theorem) *For any $P \subseteq \mathbb{R}^d$ with $d+2$ points there exists a partition (P_1, P_2) of P such that $\mathrm{conv}(P_1) \cap \mathrm{conv}(P_2) \neq \emptyset$.*

As mentioned above, Tverberg [18] generalized this theorem for larger point sets.

Theorem 1.3 (Tverberg's theorem) *Any set $P \subseteq \mathbb{R}^d$ with $n = (r-1)(d+1)+1$ points can be partitioned into r sets $P_1, \ldots, P_r$ such that $\bigcap_{i=1}^{r} \operatorname{conv}(P_i) \neq \emptyset$.*

Let P be a set of n points in $\mathbb{R}^d$. We say that $x \in \mathbb{R}^d$ has *Tverberg depth* r (with respect to P) if there is a partition of P into sets $P_1, \ldots, P_r$ such that $x \in \bigcap_{i=1}^{r} \operatorname{conv}(P_i)$. Tverberg's theorem thus states that, for any set P in $\mathbb{R}^d$, there is a point of Tverberg depth at least $\lfloor (n-1)/(d+1)+1 \rfloor = \lceil n/(d+1) \rceil$. Note that every point with Tverberg depth r also has halfspace depth r. Thus, from now on we will use the term *depth* as a shorthand for Tverberg depth. As remarked above, Tverberg's theorem immediately implies the famous centerpoint theorem [11]:

Theorem 1.4 (Centerpoint theorem) *For any set P of n points in $\mathbb{R}^d$ there is a point c such that all halfspaces containing c contain at least $\lceil n/(d+1) \rceil$ points from P.*

Finally, another classic theorem will be useful for us.

Theorem 1.5 (Carathéodory's theorem) *Suppose that P is a set of n points in $\mathbb{R}^d$ and $x \in \operatorname{conv}(P)$. Then there is a set of $d+1$ points $P' \subseteq P$ such that $x \in \operatorname{conv}(P')$.*

This means that, in order to describe a Tverberg partition of depth r, we need only $r(d+1)$ points from P. This observation is also used by Miller and Sheehy [13]. They further note that it takes $O(d^3)$ time to replace $d+2$ points by $d+1$ points using Gaussian elimination. We denote the process of replacing larger sets by sets of size $d+1$ as *pruning*, see Lemma 2.2.

1.2 Our Contribution

We now describe our results in more detail. In Sect. 2, we present a simple lifting argument which leads to an easy Tverberg approximation algorithm.

Theorem 1.6 *Let P be a set of n points in $\mathbb{R}^d$ in general position. One can compute a Tverberg point of depth $\lceil n/2^d \rceil$ for P and the corresponding partition in time $d^{O(1)}n$.*

While this does not yet give a good approximation ratio (though constant for any fixed d), it is a natural approach to the problem: it computes a higher dimensional Tverberg point via successive median partitions—just as a Tverberg point is a higher dimensional generalization of the 1-dimensional median.

By collecting several low-depth points and afterwards applying the brute-force algorithm on small point sets, we get an even higher depth in linear time for any *fixed* dimension:

Theorem 1.7 *Let P be a set of n points in $\mathbb{R}^d$. Then one can find a Tverberg point of depth $\lceil n/2(d+1)^2 \rceil$ and a corresponding partition in time $f(2^{d+1}) + d^{O(1)}n$, where $f(m)$ is the time to compute a Tverberg point of depth $\lceil m/(d+1) \rceil$ for m points by brute force.*

Finally, by combining our approach with that of Miller and Sheehy, we can improve the running time to be quasipolynomial in d:

Theorem 1.8 *Let P be a set of n points in $\mathbb{R}^d$. Then one can compute a Tverberg point of depth $\lceil n/4(d+1)^3 \rceil$ and a corresponding pruned partition in time $d^{O(\log d)}n$.*

In Sect. 4, we compare these results to the Miller–Sheehy algorithm and its extensions.

2 A Simple Fixed-Parameter Algorithm

We now present a simple algorithm that runs in linear time for any fixed dimension and computes a point of depth $\lceil n/2^d \rceil$. For this, we show how to compute a Tverberg point by recursion on the dimension. As a byproduct, we obtain a quick proof of a weaker version of Tverberg's theorem. First, however, we give a few more details about the basic operations performed by our algorithm.

2.1 Basic Operations

Our algorithm builds a Tverberg partition for a d-dimensional point set P by recursion on the dimension. In each step, we store a Tverberg partition for some point set, together with an approximate Tverberg point c. We have for each set P_i in the partition a convex combination that witnesses $c \in \operatorname{conv}(P_i)$. All the points that arise during our algorithms are obtained by repeatedly taking convex combinations of the input points, so the following simple observation lets us maintain this invariant.

Observation 2.1 *If $x_i = \sum_{p \in P_i} \alpha_p p$ and $y = \sum_i \beta_i x_i$ are convex combinations, then*

$$y = \sum_i \sum_{p \in P_i} \beta_i \alpha_p p$$

is a convex combination of the set $\bigcup P_i$ for y. □

By Carathéodory's theorem (Theorem 1.5), a Tverberg partition of depth r can be described by $r(d+1)$ points from P. In order to achieve running time $O(n)$, we need the following observation, also used by Miller and Sheehy [13].

Lemma 2.2 *Let $Q \subseteq \mathbb{R}^d$ be a set of $m \geq d+2$ points with $c \in \operatorname{conv}(Q)$, and suppose we have a convex combination of Q for c. Then we can find a subset $Q' \subset Q$ with $d+1$ points such that $c \in \operatorname{conv}(Q')$, together with a corresponding convex combination, in time $O(d^3 m)$.*

Proof Miller and Sheehy observe that replacing $d+2$ points by $d+1$ points takes $O(d^3)$ time by finding an affine dependency through Gaussian elimination, see Grötschel et al. [8, Chap. 1]. The choice of affine dependencies does not matter. Thus, in order to eliminate a point from Q, we can take any subset of size $d+2$, resolve one of the affine dependencies, and update the convex combination accordingly. Repeating this process, we can replace m points by $d+1$ points in time $(m-(d+1))O(d^3) = O(d^3 m)$. □

The process in Lemma 2.2 is called *pruning*, and we call a partition of a d-dimensional point set in which all sets have size at most $d+1$ a *pruned partition*. This will enable us to bound the cost of many operations in terms of the dimension d, instead of the number of points n.

2.2 The Lifting Argument and a Simple Algorithm

Let P be a d-dimensional point set. As a Tverberg point is a higher-dimensional version of the median, a natural way to compute a Tverberg point for P is to first project P to some lower-dimensional space, then to recursively compute a good Tverberg point for this projection, and to use this point to find a solution in the higher-dimensional space. Surprisingly, we are not aware of any such argument having appeared in the literature so far.

In what follows, we will describe how to *lift* a lower-dimensional Tverberg point into some higher dimension. Unfortunately, this process will come at the cost of a decreased depth for the lifted Tverberg point. For clarity of presentation, we first explain the lifting lemma in its simplest form. In Sect. 3.1, we then state the lemma in its full generality.

Lemma 2.3 *Let P be a set of n points in $\mathbb{R}^d$, and let h be a hyperplane in $\mathbb{R}^d$. Let $c' \in h$ be a Tverberg point of depth r for the projection of P onto h, with pruned partition $P_1, \ldots, P_r$. Then we can find a Tverberg point $c \in \mathbb{R}^d$ of depth $\lceil r/2 \rceil$ for P and a corresponding Tverberg partition in time $O(dn)$.*

Proof For every point $p \in P$, let $\mathrm{pr}(p)$ denote the projection of p onto h, and for every $Q \subseteq P$, let $\mathrm{pr}(Q)$ be the projections of all the points in Q. Let $P_1, \ldots, P_r \subseteq P$ such that $\mathrm{pr}(P_1), \ldots, \mathrm{pr}(P_r)$ is a pruned partition for $\mathrm{pr}(P)$ with Tverberg point c'. Let ℓ be the line through c' orthogonal to h.

Since our assumption implies $c' \in \mathrm{conv}(\mathrm{pr}(P_i))$ for $i = 1, \ldots, r$, it follows that ℓ intersects each $\mathrm{conv}(P_i)$ at some point $x_i \in \mathbb{R}^d$. More precisely, as we have a convex combination $c' = \sum_{p \in P_i} \alpha_p \mathrm{pr}(p)$ for each P_i, we simply get $x_i = \sum_{p \in P_i} \alpha_p p$.

Assuming an appropriate numbering, let $\widehat{Q}_i = \{x_{2i-1}, x_{2i}\}$, $i = 1, \ldots, \lceil r/2 \rceil$, be a Tverberg partition of $x_1, \ldots, x_r$. (If r is odd, the set $\widehat{Q}_{\lceil r/2 \rceil}$ contains only one point, the median.) Since the points x_i lie on the line ℓ, such a Tverberg partition exists and can be computed in time $O(r)$ by finding the median c, i.e., the element of rank $\lceil r/2 \rceil$, according to the order along ℓ [5, Chap. 9]. We claim that c is a Tverberg point for P of depth $\lceil r/2 \rceil$. Indeed, we have

$$c \in \mathrm{conv}(\widehat{Q}_i) = \mathrm{conv}(\{x_{2i-1}, x_{2i}\}) \subseteq \mathrm{conv}(P_{2i-1} \cup P_{2i})$$

for $1 \leq i \leq \lceil r/2 \rceil$. Thus, if we set $Q_i := P_{2i-1} \cup P_{2i}$, then $Q_1, \ldots, Q_{\lceil r/2 \rceil}$ is a Tverberg partition for the point c. The total time to compute c and the Q_i is $O(n)$, as claimed. See Fig. 2 for a two-dimensional illustration of the lifting argument. □

Theorem 1.6 is now a direct consequence of Lemma 2.3.

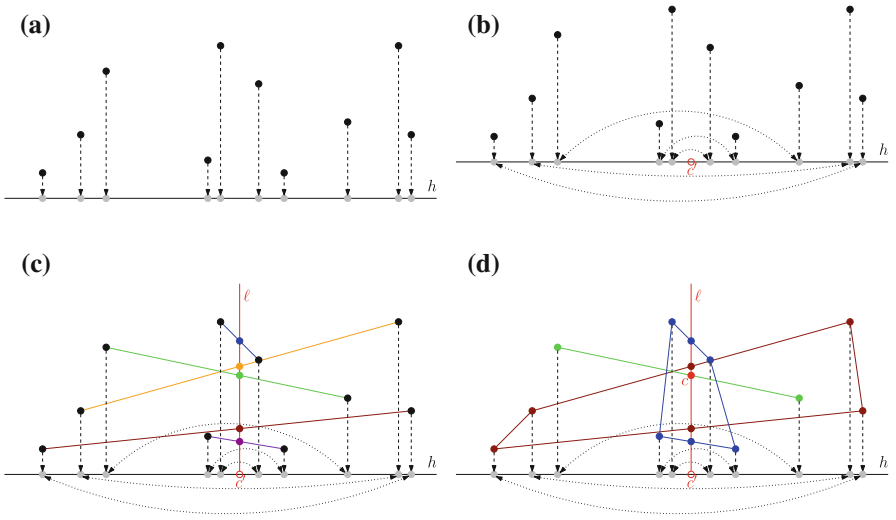

Fig. 2 Illustrating the lifting lemma in the plane: we project the point set P to the line h and find a Tverberg partition and a Tverberg point c' for the projection. Then we construct the line ℓ through c' that is perpendicular to h, and we take the intersection with the lifted convex hulls of the Tverberg partition. We then find the median c and the corresponding partition for the intersections along ℓ. Finally, we group the points according to this partition. **a** Project, **b** find partition, **c** intersect hulls of the sets with $h^{\perp}$, and **d** find median of intersections and combine (Color figure online)

Theorem 2.4 (Theorem 1.6, restated) *Let P be a set of n points in $\mathbb{R}^d$ in general position. One can compute a Tverberg point of depth $\lceil n/2^d \rceil$ for P and the corresponding partition in time $d^{O(1)}n$.*

Proof If $d = 1$, we obtain a Tverberg point and a corresponding partition by finding the median c of P [5] and pairing each point to the left of c with exactly one point to the right of c.

If $d > 1$, we project P onto the hyperplane $x_d = 0$. This gives an n-point set $P' \subseteq \mathbb{R}^{d-1}$. We recursively find a point of depth $\lceil n/2^{d-1} \rceil$ and a corresponding pruned partition for P'. We then apply Lemma 2.3 to get a point $c \in \mathbb{R}^d$ of depth $r = \lceil \lceil n/2^{d-1} \rceil /2 \rceil \geq \lceil n/2^d \rceil$ for P, together with a partition. Each set has at most $2d$ points, so by applying Lemma 2.2 to each set, it takes $O(d^4 r)$ time to prune all sets.

This yields a total running time of $T_d(n) \leq T_{d-1}(n) + d^{O(1)}n$, which implies the result. □

In particular, Theorem 1.6 gives a weak version of Tverberg's theorem with a simple proof.

Corollary 2.5 (Weak Tverberg theorem) *Let P be a set of n points in $\mathbb{R}^d$. Then P can be partitioned into $\lceil n/2^d \rceil$ sets $P_1, \ldots, P_{\lceil n/2^d \rceil}$ such that*

$$\bigcap_{i=1}^{\lceil n/2^d \rceil} \operatorname{conv}(P_i) \neq \emptyset.$$

2.3 An Improved Approximation Factor

In order to improve the approximation factor, we will now use an easy bootstrapping approach. A Tverberg partition of depth r in $\mathbb{R}^d$ needs only $(d+1)r$ points. This means that after finding a point of depth $n/2^d$, we still have $n\left(1-(d+1)/2^d\right)$ unused points at our disposal. The next lemma shows how to leverage these points to achieve an even higher Tverberg depth.

Lemma 2.6 *Let $\rho \geq 2$ and $q(m,d)$ be a function such that for any m-point set $Q \subseteq \mathbb{R}^d$ we can compute a point of depth $\lceil m/\rho \rceil$ and a corresponding pruned partition in time $q(m,d)$.*

Let $P \subseteq \mathbb{R}^d$ with $|P| = n$, and let $\beta \in [2, n/\rho]$ be a constant. Define the target depth δ as $\delta := \lceil n/\beta\rho \rceil$. Then we can find $\alpha := \lceil \frac{n(1-1/\beta)}{\delta(d+1)} \rceil$ disjoint subsets $Q_1, \ldots, Q_\alpha$ of P such that for each Q_i we have a Tverberg point c_i of depth δ and a pruned partition $\mathcal{Q}_i$. This takes total time

$$O\Big(\frac{(\beta-1)\rho\, q(n,d)}{d+1}\Big).$$

Proof Let $P_1 := P$. We take an arbitrary subset $P_1' \subseteq P_1$ with $\lceil n/\beta \rceil$ points and find a Tverberg point c_1 of depth δ and a corresponding pruned partition $\mathcal{Q}_1$ for P_1'. This takes time $q(n,d)$, and the set $Q_1 := \bigcup_{Z \in \mathcal{P}_1} Z$ contains at most $\delta(d+1)$ points. Set $P_2 := P_1 \setminus Q_1$ and repeat. The resulting sets Q_i are pairwise disjoint, and we can repeat this process until

$$n - i\delta(d+1) < \frac{n}{\beta}.$$

This gives

$$\alpha \geq i > \left\lceil \frac{n(1-1/\beta)}{\delta(d+1)} \right\rceil.$$

Thus, we obtain α points $c_1, \ldots, c_\alpha$ with corresponding Tverberg partitions $\mathcal{Q}_1, \ldots, \mathcal{Q}_\alpha$, each of depth at least $\lceil n/\beta\rho \rceil$, as desired. □

For example, by Theorem 1.6 we can find a point of depth $\lceil n/2^d \rceil$ and a corresponding pruned partition in time $d^{O(1)}n$. Thus, by applying Lemma 2.6 with $c = 2$, $\rho = 2^d$, we can also find $\lceil n/(2\lceil n/2^{d+1} \rceil (d+1)) \rceil \approx 2^d/(d+1)$ points of depth $\lceil n/2^{d+1} \rceil$ in linear time.

In order to make use of Lemma 2.6, we will also need a lemma that describes how we can combine these points in order to increase the total depth. This generalizes a similar lemma by Miller and Sheehy [13, Lemma 4.1].

Lemma 2.7 *Let P be a set of n points in $\mathbb{R}^d$, and let $P = \biguplus_{i=1}^{\alpha} P_i$ be a partition of P. Furthermore, suppose that for each P_i we have a Tverberg point $c_i \in \mathbb{R}^d$ of depth r, together with a corresponding pruned partition $\mathcal{P}_i$. Let $C := \{c_i \mid 1 \leq i \leq \alpha\}$ and c be a point of depth r' for C, with corresponding pruned partition $\mathcal{C}$. Then c is a point*

of depth rr' for P. Furthermore, we can find a corresponding pruned partition in time $d^{O(1)}n$.

Proof For $i = 1, \ldots, \alpha$, write $\mathcal{P}_i = \{Q_{i1}, \ldots, Q_{ir}\}$, and write $\mathcal{C} = \{D_1, \ldots, D_{r'}\}$. For $a = 1, \ldots, r', b = 1, \ldots, r$, we define sets Z_{ab} as

$$Z_{ab} := \bigcup_{c_i \in D_a} Q_{ib}.$$

We claim that the set $\mathcal{Z} := \{Z_{ab} \mid a = 1, \ldots, r'; b = 1, \ldots, r\}$ is a Tverberg partition of depth rr' for P with Tverberg point c. By definition, $\mathcal{Z}$ is a partition with the appropriate number of elements. It only remains to check that $c \in \operatorname{conv}(Z_{ab})$ for each Z_{ab}. Indeed, we have

$$\begin{aligned} c \in \operatorname{conv}(D_a) &= \operatorname{conv}\Big(\bigcup_{c_i \in D_a} \{c_i\} \Big) \subseteq \operatorname{conv}\Big(\bigcup_{c_i \in D_a} \operatorname{conv}(Q_{ib}) \Big) \\ &= \operatorname{conv}\Big(\bigcup_{c_i \in D_a} Q_{ib} \Big) = \operatorname{conv}(Z_{ab}) \end{aligned}$$

for $a = 1 \ldots r', b = 1 \ldots r$.

As the partitions $\mathcal{P}_i$ and $\mathcal{C}$ were pruned, each Z_{ab} consists of at most $(d+1)^2$ points. Thus, by Lemma 2.2, each Z_{ab} can be pruned in time $O(d^5)$. Since $|\mathcal{Z}| \leq n$, the lemma follows. □

Combining Lemmas 2.6 and 2.7, we can now prove Theorem 1.7.

Theorem 2.8 (Theorem 1.7, restated) *Let P be a set of n points in $\mathbb{R}^d$. Then one can find a Tverberg point of depth $\lceil n/2(d+1)^2 \rceil$ and a corresponding partition in time $f(2^{d+1}) + d^{O(1)}n$, where $f(m)$ is the time to compute a Tverberg point of depth $\lceil m/(d+1) \rceil$ for m points by brute force, together with an associated Tverberg partition.*

Proof If $n \leq 2^{d+1}$, we solve the problem by brute-force in $f(2^{d+1})$ time. Otherwise, we apply Lemma 2.6 with $c = 2$ and $\rho = 2^d$ to obtain a set C of

$$|C| = \left\lceil \frac{n}{2\lceil n/2^{d+1} \rceil (d+1)} \right\rceil$$

points of depth $\lceil n/2^{d+1} \rceil$ for P with corresponding pruned partitions in time $d^{O(1)}n$. We then use the brute-force algorithm to get a Tverberg point for C with depth $\lceil |C|/(d+1) \rceil$ and a corresponding partition, in time $f(|C|)$. Finally, we apply Lemma 2.7 to obtain a Tverberg point and corresponding partition in time $d^{O(1)}n$. Using that $\lceil a \lceil b \rceil \rceil \geq \lceil ab \rceil$ and $\lceil a \rceil \lceil b \rceil \geq \lceil a \lceil b \rceil \rceil$ for $a, b \geq 0$, we get that the resulting depth is

$$\left\lceil \frac{n}{2^{d+1}} \right\rceil \cdot \left\lceil \frac{|C|}{d+1} \right\rceil \geq \left\lceil \left\lceil \frac{n}{2^{d+1}} \right\rceil \frac{n}{2\lceil n/2^{d+1} \rceil (d+1)^2} \right\rceil = \left\lceil \frac{n}{2(d+1)^2} \right\rceil,$$

and the total running time is $f(2^{d+1}) + d^{O(1)}n$, as desired. □

Instead of brute force, we can also use the algorithm by Miller and Sheehy to find a point among the deep points. This gives a worse depth, but it is slightly faster.

Theorem 2.9 *Let P be a set of n points in $\mathbb{R}^d$. Then one can compute a Tverberg point of depth $\lceil n/4(d+1)^3 \rceil$ and a corresponding partition in time $2^{O(d \log d)} + d^{O(1)}n$.*

Proof For $n \leq 2^{d+1}$ we use the Miller–Sheehy algorithm to get a point of depth $\lceil n/2(d+1)^2 \rceil$ in time $2^{O(d \log d)}$. Otherwise, we proceed as in the proof of Theorem 1.7 to obtain a set C of

$$|C| = \left\lceil \frac{n}{2\lceil n/2^{d+1} \rceil (d+1)} \right\rceil$$

Tverberg points of depth $\lceil n/2^{d+1} \rceil$ and corresponding pruned partitions in time $d^{O(1)}n$. The Miller–Sheehy algorithm then gives a Tverberg point for C of depth $\lceil |C|/2(d+1)^2 \rceil$ in time $|C|^{O(\log d)} = 2^{O(d \log d)}$. Finally, we apply Lemma 2.7. This takes time $d^{O(1)}n$ and yields a Tverberg point and pruned partition of depth

$$\left\lceil \frac{n}{2^{d+1}} \right\rceil \cdot \left\lceil \frac{|C|}{2(d+1)^2} \right\rceil \geq \left\lceil \left\lceil \frac{n}{2^{d+1}} \right\rceil \frac{n}{4\lceil n/2^{d+1} \rceil (d+1)^3} \right\rceil = \left\lceil \frac{n}{4(d+1)^3} \right\rceil,$$

as claimed. □

3 An Improved Running Time

The algorithm from the previous section runs in linear time for any fixed dimension, but the constants are huge. In this section, we show how to speed up our approach through an improved recursion, and we obtain an algorithm with running time $d^{O(\log d)}n$ while losing a depth factor of $1/2(d+1)$.

3.1 A More General Version of the Lifting Argument

We first present a more general version of the lifting argument in Lemma 2.3. For this, we need some more notation. Let $P \subseteq \mathbb{R}^d$ be finite. A k-dimensional *flat* $F \subseteq \mathbb{R}^d$ (often abbreviated as *k-flat*) is defined as a k-dimensional affine subspace of $\mathbb{R}^d$ (or, equivalently, as the affine hull of $k+1$ affinely independent points in $\mathbb{R}^d$). We call a k-dimensional flat $F \subseteq \mathbb{R}^d$ a *Tverberg k-flat of depth r for P* if there is a partition of P into sets $P_1, \ldots, P_r$ such that $\text{conv}(P_i) \cap F \neq \emptyset$ for all $i = 1, \ldots, r$. This generalizes the notion of a Tverberg point.

Lemma 3.1 *Let P be a set of n points in $\mathbb{R}^d$, and let $h \subseteq \mathbb{R}^d$ be a k-flat. Suppose we have a Tverberg point $c \in h$ of depth r for $pr(P) := pr_h(P)$, as well as a corresponding Tverberg partition. Let $h_c^\perp$ be the $(d-k)$-flat orthogonal to h that passes through c. Then $h_c^\perp$ is a Tverberg $(d-k)$-flat for P of depth r, with the same Tverberg partition.*

Proof Let $\mathrm{pr}(P_1), \ldots, \mathrm{pr}(P_r)$ be the Tverberg partition for the projection $\mathrm{pr}(P)$. It suffices to show that $\mathrm{conv}(P_i)$ intersects $h_c^\perp$ for $i = 1, \ldots, r$. Indeed, for $P_i = \{p_{i1}, \ldots, p_{il_i}\}$ let $c = \sum_{j=1}^{l_i} \lambda_j \mathrm{pr}(p_{ij})$ be a convex combination that witnesses $c \in \mathrm{conv}(\mathrm{pr}(P_i))$. We now write each $p_{ij} = \mathrm{pr}(p_{ij}) + \mathrm{pr}^\perp(p_{ij})$, where $\mathrm{pr}^\perp(\cdot)$ denotes the projection onto the orthogonal complement $h^\perp$ of h. Then

$$\sum_{j=1}^{l_i} \lambda_j p_{ij} = \sum_{j=1}^{l_i} \lambda_j \mathrm{pr}(p_{ij}) + \sum_{j=1}^{l_i} \lambda_j \mathrm{pr}^\perp(p_{ij}) \in c + h^\perp = h_c^\perp,$$

as claimed. □

Lemma 3.1 lets us use a good algorithm for *any fixed dimension* to improve the general case.

Lemma 3.2 *Let $\delta \geq 1$ be a fixed integer. Suppose we have an algorithm $\mathcal{A}$ with the following property: for every point set $Q \subseteq \mathbb{R}^\delta$, the algorithm $\mathcal{A}$ constructs a Tverberg point of depth $\lceil |Q|/\rho \rceil$ for Q as well as a corresponding pruned partition in time $f(|Q|)$.*

Then, for any n-point set $P \subseteq \mathbb{R}^d$ and for any $d \geq \delta$, we can find a Tverberg point of depth $\lceil n/\rho^{\lceil d/\delta \rceil} \rceil$ and a corresponding pruned partition in time $\lceil d/\delta \rceil f(n) + d^{O(1)} n$.

Proof Set $k := \lceil d/\delta \rceil$. We use induction on k to show that such an algorithm exists with running time $k(f(n) + d^{O(1)} n)$. If $k = 1$, we can just use algorithm $\mathcal{A}$, and there is nothing to show.

Now suppose $k > 1$. Let $h \subseteq \mathbb{R}^d$ be a δ-flat in $\mathbb{R}^d$, and let $\mathrm{pr}(P)$ be the projection of P onto h. We use algorithm $\mathcal{A}$ to find a Tverberg point c of depth $\lceil n/\rho \rceil$ for $\mathrm{pr}(P)$ as well as a corresponding pruned partition $\mathrm{pr}(P_1), \ldots, \mathrm{pr}(P_{\lceil n/\rho \rceil})$. This takes time $f(n)$. By Lemma 3.1, the $(d-\delta)$-flat $h_c^\perp$ is a Tverberg flat of depth $\lceil n/\rho \rceil$ for P, with corresponding pruned partition $P_1, \ldots, P_{\lceil n/\rho \rceil}$. For each i, we can thus find a point q_i in $\mathrm{conv}(P_i) \cap h_c^\perp$ in time $d^{O(1)}$.

Now consider the point set $Q = \{q_1, \ldots, q_{\lceil n/\rho \rceil}\} \subseteq h_c^\perp$. The set Q is $(d-\delta)$-dimensional. Since $\lceil (d-\delta)/\delta \rceil = k - 1$, we can inductively find a Tverberg point c' for Q of depth $\lceil |Q|/\rho^{\lceil d/\delta \rceil - 1} \rceil \geq \lceil n/\rho^{\lceil d/\delta \rceil} \rceil$ and a corresponding pruned partition $\mathcal{Q}$ in total time $(k-1)(f(n) + d^{O(1)} n)$. Now, c' is a Tverberg point of depth $n/\rho^{\lceil d/\delta \rceil}$ for P: a corresponding Tverberg partition is obtained by replacing each point q_i in the partition $\mathcal{Q}$ by the corresponding subset P_i. The resulting partition can be pruned in time $d^{O(1)} n$. Thus, the total running time is

$$(k-1)(f(n) + d^{O(1)} n) + f(n) + d^{O(1)} n = k(f(n) + d^{O(1)} n),$$

and since $k = O(d)$, the claim follows. □

For example, the result of Agarwal et al. [1] gives a point of depth $\lceil n/4 \rceil$ in 3 dimensions in time $O(n \log n)$. Thus, we can find a point of depth $n/4^{\lceil d/3 \rceil}$ in time $O(n \log n + d^{O(1)} n)$.

3.2 An Improved Algorithm

Finally, we show how to combine the above techniques to obtain an algorithm with a better running time. The idea is as follows: using Lemma 3.2, we can reduce one d-dimensional instance to two instances of dimension $d/2$. We would like to proceed recursively, but unfortunately, this reduces the depth of the partition. To fix this, we apply Lemmas 2.6, 2.7 and the Miller–Sheehy algorithm.

Theorem 3.3 (Theorem 1.8, restated) *Let P be a set of n points in $\mathbb{R}^d$. Then one can compute a Tverberg point of depth $\lceil n/4(d+1)^3 \rceil$ and a corresponding pruned partition in time $d^{O(\log d)}n$.*

Proof We prove the theorem by induction on d. As usual, for $d = 1$ the problem reduces to median computation, and the result is immediate.

Now let $d \geq 2$. By induction, for any at most $\lceil d/2 \rceil$-dimensional point set $Q \subseteq \mathbb{R}^{\lceil d/2 \rceil}$ there exists an algorithm that returns a Tverberg point of depth $\lceil |Q|/4(\lceil d/2 \rceil + 1)^3 \rceil$ and a corresponding pruned partition in time $d^{\alpha \log \lceil d/2 \rceil}n$, for some sufficiently large constant $\alpha > 0$.

Thus, by Lemma 3.2 (with $\delta = \lceil d/2 \rceil$), there exists an algorithm that can compute a Tverberg point for P of depth $\lceil n/16(\lceil d/2 \rceil + 1)^6 \rceil$ and a corresponding Tverberg partition in total time $2d^{\alpha \log \lceil d/2 \rceil} + d^{O(1)}n$. Now we apply Lemma 2.6 with $c = 2$ and $\rho = 16(\lceil d/2 \rceil + 1)^6$. The lemma shows that we can compute a set C of $\lceil 16(\lceil d/2 \rceil + 1)^6/(d+1) \rceil$ points of depth $\delta = \lceil n/32(\lceil d/2 \rceil + 1)^6 \rceil$ and corresponding (disjoint) pruned partitions in time $d^{\alpha \log \lceil d/2 \rceil + O(1)}n$. Applying the Miller–Sheehy algorithm, we can find a Tverberg point for C of depth $\lceil |C|/2(d+1)^2 \rceil$ and a corresponding pruned partition in time $|C|^{O(\log d)}$. Now, Lemma 2.7 shows that in additional $d^{O(1)}n$ time, we obtain a Tverberg point and a corresponding Tverberg partition for P of size

$$\left\lceil \frac{n}{2 \cdot 16(\lceil d/2 \rceil + 1)^6} \right\rceil \left\lceil \frac{16(\lceil d/2 \rceil + 1)^6}{2(d+1)^2(d+1)} \right\rceil \geq \left\lceil \frac{n}{4(d+1)^3} \right\rceil,$$

since $\lceil a \rceil \lceil b \rceil \geq \lceil ab \rceil$ for all $a, b \geq 0$.

It remains to analyze the running time. Adding the various terms, we obtain a time bound of

$$T(n,d) = d^{\alpha \log \lceil d/2 \rceil + O(1)}n + |C|^{O(\log d)} + d^{O(1)}n.$$

Since $|C| = d^{O(1)}$, using $\log \lceil d/2 \rceil \leq \log(d/2) + \log(2\lceil d/2 \rceil/d) \leq \log(d/2) + \log(4/3)$, we get

$$\begin{aligned} T(n,d) &\leq d^{\alpha \log \lceil d/2 \rceil + O(1)}n + d^{O(\log d)}n \\ &\leq d^{\alpha \log d - \alpha/2}n + d^{\beta \log d}n, \end{aligned}$$

Table 1 Comparing our results to Miller–Sheehy and extensions

Algorithm	Running time	Depth
Theorem 1.6	$O(n)$	$n/2^d$
Miller–Sheehy	$n^{O(\log d)}$	$n/2(d+1)^2$
Theorem 1.7	$O\big(f(2^d)+d^{O(1)}n\big)$	$n/2(d+1)^2$
Miller–Sheehy generalized ($r=d+1$)	$O\big(f(d)n^2\big)$	$\approx n/(d+1)^3$
Theorem 2.9	$O\big(2^{O(d\log d)}+n\big)$	$n/4(d+1)^3$
Miller–Sheehy bootstrapped	$d^{O(\log d)}n^3$	$\approx n/2(d+1)^4$
Theorem 1.8	$d^{O(\log d)}n$	$n/4(d+1)^3$

for α large enough and some $\beta > 0$, independent of d. Hence, for large enough α we have

$$T(n,d) \le d^{\alpha \log d} n = d^{O(\log d)} n,$$

as claimed. This completes the proof. □

Thus, we can compute a polynomial approximation to a Tverberg point in time pseudopolynomial in d and linear in n.

4 Comparison to Miller–Sheehy

In Table 1, we give a more detailed comparison of our results to the Miller–Sheehy algorithm and its extensions. In Sect. 5.2 of their paper, Miller and Sheehy describe a generalization of their approach that improves the running time for small d by computing higher order Tverberg points of depth r by brute force. The approximation quality deteriorates by a factor of $r/2$. No exact bounds are given, but as far as we can tell, one can achieve a running time of $O(f(d)n^2)$ for fixed d by setting the parameter $r = d+1$, while losing a factor of $(d+1)/2$ in the approximation.

Furthermore, even though it is not explicitly mentioned in their paper, we think that it is possible to also bootstrap the Miller–Sheehy algorithm (for a better running time in terms of d, while losing another factor of $(d+1)$ in the output). This is done by performing the generalized procedure [13, Sect. 5.2] with $r = d+1$, but using the original Miller–Sheehy algorithm instead of the brute-force algorithm. Table 1 shows a rough comparison (ceilings omitted) of the different approaches. Again, f denotes the running time of the brute force algorithm.

We should emphasize that for all dimensions d with $2^d \leq 2(d+1)^2$, i.e., $d \leq 7$, our simplest algorithm outperforms every other approximation algorithm in both running time and approximation ratio. For example, it gives a $1/2$-approximate Tverberg point in 3 dimensions in linear time.

5 Conclusion and Outlook

We have presented a simple algorithm for finding an approximate Tverberg point. It runs in linear time for any fixed dimension. Using more sophisticated tools and combining our methods with known results, we managed to improve the running time to $d^{O(\log d)}n$, while getting within a factor of $1/4(d+1)^2$ of the bound from Tverberg's theorem. Unfortunately, the resulting running time remains quasipolynomial in d, and we still do not know whether there exists a polynomial algorithm (in n and d) for finding an approximate Tverberg point of linear depth.

However, we are hopeful that our techniques constitute a further step towards a truly polynomial time algorithm and that such an algorithm will eventually be discovered—maybe even by a more clever combination of our algorithm with that of Miller and Sheehy. An alternative promising approach, suggested to us by Don Sheehy, derives from a beautiful proof of Tverberg's theorem. It is due to Sarkaria and can be found in Matousek's book [11, Chap. 8]. It uses the colorful Carathéodory theorem:

Theorem 5.1 (Colorful Carathéodory) *Let* $C_1 \uplus \cdots \uplus C_{d+1} \subseteq \mathbb{R}^d$ *such that for* $i = 1, \ldots, d+1$, *we have* $0 \in \operatorname{conv}(C_i)$. *Then there is a set* C *of* $d+1$ *points with* $0 \in \operatorname{conv}(C)$ *and* $|C_i \cap C| = 1$.

Sarkaria's proof transforms a d-dimensional instance of n points of the Tverberg point problem to a Colorful Carathéodory problem in approximately dn dimensions.

The question now is whether such a colorful simplex can be found in time polynomial in both d and n. This would lead to a polynomial time algorithm for computing a Tverberg point. Observe that this would not contradict any complexity theoretic assumptions: an algorithm that *finds* such a point does not necessarily have to decide whether a given point indeed is a Tverberg point.

The simplest proof of Colorful Carathéodory leads directly to an algorithm for finding such a colorful simplex and works as follows: take an arbitrary colorful simplex. If the origin is not contained in it, delete the farthest color and take a point of that color that together with the other points induces a simplex that is closer to the origin. It is unknown whether this procedure runs in polynomial time for both d and n. Settling this question would constitute major progress on the problem (see [12,16] for work in this direction).

Yet another approach would be to relax Sarkaria's proof and to try to formulate it as an approximation problem, which might be easier to solve. However, it is not clear how to state such an approximation to the Colorful Carathéodory problem in a way that leads to an approximate Tverberg point. Perhaps via such a method, our algorithms can be improved further.

It is known that the problem of deciding whether a given point has at least a certain depth is NP-complete [17]. It is possible to strengthen this result to show that in

$\mathbb{R}^{d+1}$, the problem is d-SUM hard, using the approach by Knauer et al. [10]. However, this does not tell us anything about the actual problem of computing a point of depth $n/(d+1)$. Such a point is guaranteed to exist, so it is not clear how to prove the problem hard using "standard" NP-completeness theory. Rather, we think that a hardness proof along the lines of complexity classes such as PPAD or PLS [14] should be pursued.

Finally, a common issue with Tverberg point (and centerpoint) algorithms in high dimensions, also pointed out by Clarkson et al. [4], is that the coefficients arising during the algorithm might become exponentially large. While this is not a problem in our uniform cost model, for implementations of the algorithm it seems necessary to bound these. In particular, it would be interesting to investigate the bit complexity of the intermediate solutions arising during the pruning process. As an alternative approach, one might try to perturb the points in the process, thereby lowering the precision of the coefficients. Additionally, one might have to introduce a notion of *almost approximate Tverberg points*, where the point that is returned does not have to lie *inside* all sets, but only *close to* them.

Acknowledgments We would like to thank Nabil Mustafa for suggesting the problem to us. We also thank him and Don Sheehy for helpful discussions and insightful suggestions. We would further like to thank the anonymous referees for their helpful and detailed comments. Werner was funded by Deutsche Forschungsgemeinschaft within the Research Training Group (Graduiertenkolleg) "Methods for Discrete Structures."

References

1. Agarwal, P.K., Sharir, M., Welzl, E.: Algorithms for center and Tverberg points. ACM Trans. Algorithms 5(1), Art. 5 (2009)
2. Chan, T.M.: An optimal randomized algorithm for maximum Tukey depth. In: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 430–436 (2004)
3. Chazelle, B.: The Discrepancy Method: Randomness and Complexity. Cambridge University Press, Cambridge, MA (2000)
4. Clarkson, K.L., Eppstein, D., Miller, G.L., Sturtivant, C., Teng, S.-H.: Approximating center points with iterated Radon points. Int. J. Comput. Geom. Appl. **6**(3), 357–377 (1996)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd edn. MIT Press, Cambridge (2009)
6. Danzer, L., Grünbaum, B., Klee, V.: Helly's theorem and its relatives. In: Proceedings of the Symposium on Pure Mathematics, vol. VII, pp. 101–180. American Mathematical Society, Providence (1963)
7. Edelsbrunner, H.: Algorithms in Combinatorial Geometry. Springer, Berlin (1987)
8. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, Volume 2 of Algorithms and Combinatorics, 2nd edn. Springer, Berlin (1993)
9. Jadhav, S., Mukhopadhyay, A.: Computing a centerpoint of a finite planar set of points in linear time. Discrete Comput. Geom. **12**(3), 291–312 (1994)
10. Knauer, C, Tiwary, H.R., Werner, D.: On the computational complexity of Ham-Sandwich cuts, Helly sets, and related problems. In: 28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011), vol. 9, pp. 649–660. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Wadern (2011)
11. Matoušek, J.: Lectures on Discrete Geometry. Springer, New York (2002)
12. Meunier, F., Deza, A.: A further generalization of the colourful Carathéodory theorem. http://arxiv/abs/1107.3380 (2011)
13. Miller, G.L., Sheehy, D.R.: Approximate centerpoints with proofs. Comput. Geom. Theory Appl. **43**(8), 647–654 (2010)
14. Papadimitriou, C.H.: On the complexity of the parity argument and other inefficient proofs of existence. J. Comput. Syst. Sci. **48**(3), 498–532 (1994)

15. Rado, R.: A theorem on general measure. J. Lond. Math. Soc. **21**, 291–300 (1946)
16. Rong, G.: On algorithms for the colourful linear programming feasibility problem. McMaster University, Master's thesis (2012)
17. Teng, S.-H.: Points, spheres, and separators: a unified geometric approach to graph partitioning. PhD thesis, School of Computer Science, Carnegie Mellon University (1992)
18. Tverberg, H.: A generalization of Radon's theorem. J. Lond. Math. Soc. **41**, 123–128 (1966)