Approximating Tverberg Points in Linear Time for Any Fixed Dimension

Wolfgang Mulzer
Institut für Informatik, Freie Universität Berlin
Takustr. 9
14195 Berlin, Germany
mulzer@inf.fu-berlin.de
http://page.mi.fu-berlin.de/mulzer

Daniel Werner*
Institut für Informatik, Freie Universität Berlin
Takustr. 9
14195 Berlin, Germany
daniel.werner@fu-berlin.de
http://page.mi.fu-berlin.de/dawerner

ABSTRACT

Let $P \subseteq \mathbb{R}^d$ be a d-dimensional n-point set. A *Tverberg* partition of P is a partition of P into r sets $P_1, \ldots, P_r$ such that the convex hulls $\mathrm{conv}(P_1), \ldots, \mathrm{conv}(P_r)$ have non-empty intersection. A point in $\bigcap_{i=1}^{r} \mathrm{conv}(P_i)$ is called a *Tverberg point* of depth r for P. A classic result by Tverberg implies 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 [10].

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*

General Terms

Algorithms, Theory

Keywords

discrete geometry, Tverberg's theorem, centerpoint, approximation, high dimension

*This research was funded by Deutsche Forschungsgemeinschaft within the Research Training Group (Graduiertenkolleg) "Methods for Discrete Structures".

SCG'12, June 17–20, 2012, Chapel Hill, North Carolina, USA.

1. INTRODUCTION

Let $P \subseteq \mathbb{R}^d$ be a d-dimensional point set with n points. 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 median to the high-dimensional point set P. A very natural means to accomplish this are *centerpoints*; a point $c \in \mathbb{R}^d$ is a centerpoint for P if every halfspace that contains c meets P in at least $n/(d+1)$ points. Equivalently, for every direction $\vec{v}$, the projection of c onto $\vec{v}$ splits the projection of P onto $\vec{v}$ into pieces with no more than $dn/(d+1)$ points [3]. A classic result in discrete geometry, the centerpoint theorem, shows that there exists a centerpoint for every point set [5, 12].

However, if we actually would like to compute a centerpoint for a given point set, the situation becomes more involved. Helly's theorem implies that the set of all centerpoints is given by the intersection of $O(n^d)$ halfspaces [6], so we can find a centerpoint in $O(n^d)$ time through linear programming. Chan [1] shows how to improve this running time to $O(n^{d-1})$ steps in expectation. He actually solves the harder problem of finding a point with maximum *Tukey depth*. The Tukey depth of a point c' is defined as the minimum number of points in P that are met by any halfspace containing c'. A centerpoint has Tukey depth at least $n/(d+1)$. If the dimension is not fixed, a result by Teng shows that it is co-NP-hard to check whether a given point is a centerpoint [13]. In two dimensions, a center point can be found in linear time [7].

Since a running time of $O(n^{d-1})$ is not feasible for large d, it makes sense to look for faster approximate solutions. A classic approach uses ε-approximations [2]: in order to obtain a point of Tukey 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, using the linear-programming method. This gives the desired approximation with constant probability, and the resulting running time after the sampling step is constant. 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 [2], at the expense of a more complicated algorithm. Nonetheless, in either case the resulting running time, though constant, still grows exponentially with d, an undesirable feature for large dimensions.

This situation motivated Clarkson et al. [3] to look for more efficient randomized algorithms for approximate cen-

terpoints. They give a simple probabilistic algorithm that computes a point of Tukey depth $O(n/(d+1)^2)$ 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 a failure efficiently.

More than ten years later, Miller and Sheehy [10] launched a new attack at the problem. Their goal is to develop a deterministic algorithm for approximating 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 $P_1,\dots,P_r$ such that the convex hulls $\mathrm{conv}(P_1),\dots,\mathrm{conv}(P_r)$ intersect. Such a partition is called a *Tverberg* partition of P. Any point in $\bigcap_{i=1}^r \mathrm{conv}(P_i)$ must be a center point.

More generally, for any partition $P_1,\dots,P_{r'}$ of P into r' sets and any point c that is contained in the convex hull of each of the sets, we say that c has *Tverberg depth* r' with respect to P. Consequently, c is called an *approximate Tverberg point* (of depth r'). See Figure 1.

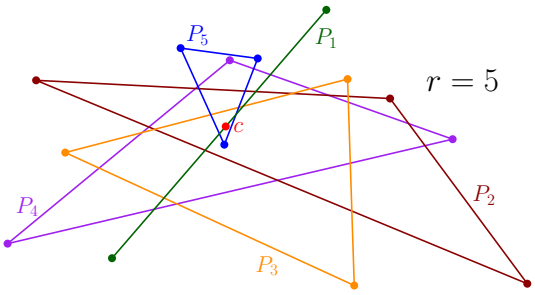


Figure 1: c is a point of Tverberg depth $r=5$.

Miller and Sheehy describe how to find disjoint subsets $Q_1,\dots,Q_{r'}$, $r'=\lceil n/2(d+1)^2\rceil$, of P and a point $c\in\mathbb{R}^d$, such that each Q_i contains $d+1$ points and such that $c\in\mathrm{conv}(Q_i)$. Hence, c constitutes an approximate centerpoint for P, and the Q_i 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 depends exponentially on $\log d$.

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 for a set of points P that contains $\lceil n/4(d+1)^3\rceil$ 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 [9]. A classic fact about convexity is Radon's theorem.

THEOREM 1.1 (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 [14] generalized this theorem for larger point sets.

THEOREM 1.2 (TVERBERG'S THEOREM). *Any set $P\subseteq\mathbb{R}^d$ with $n=(r-1)(d+1)+1$ points can be partitioned into r sets $P_1,\dots,P_r$ such that $\bigcap_{i=1}^r\mathrm{conv}(P_i)\neq\emptyset$.*

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 $P_1,\dots,P_r$ such that $x\in\bigcap_{i=1}^r\mathrm{conv}(P_i)$. 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 Tukey 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 (see [9]):

THEOREM 1.3 (CENTERPOINT THEOREM). *For any set P of n points in $\mathbb{R}^d$ there is a point c such that all half-spaces containing c contain at least $\lceil n/(d+1)\rceil$ points from P.*

Finally, another classic theorem will be useful for us.

THEOREM 1.4 (CARATHÉODORY'S THEOREM). *Suppose that P is a set of n points in $\mathbb{R}^d$ and $x\in\mathrm{conv}(P)$. Then there is a set of $d+1$ points $P'\subseteq P$ such that $x\in\mathrm{conv}(P')$.*

This means that, in order to describe a Tverberg partition of depth r, we only need $r(d+1)$ points from P. This observation is also used by Miller and Sheehy [10]. They also observe that replacing $d+2$ by $d+1$ points can be done in $O(d^3)$ time by using Gaussian elimination. We denote the process of replacing larger sets by sets of size $d+1$ as *pruning*.

1.2 Our contribution

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

THEOREM 1.5. *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 very 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 1.6. *Let P be a set of n points in $\mathbb{R}^d$. Then one can compute 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 for computing a Tverberg point of depth $\lceil m/(d+1)\rceil$ for m points brute force.*

Finally, by combining our approach with that of Miller and Sheehy, we improve our algorithm to yield an algorithm whose running time is quasipolynomial in d:

THEOREM 1.7. *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 Section 4, we compare these results to the Miller-Sheehy algorithm and its extensions.

2. A SIMPLE FIXED PARAMETER ALGORITHM

First, we 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.

2.1 The lifting argument and a simple algorithm

Let P be a d-dimensional point set. 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 use this point to find a solution in the higher-dimensional space. Surprisingly, we are not aware of any argument along these lines 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 Section 3.1, we then state the lemma in its full generality.

LEMMA 2.1. *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, and suppose we know a corresponding Tverberg partition. 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 $O(n)$.*

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 Tverberg partition for $\mathrm{pr}(P)$ with Tverberg point c'. Let ℓ be the line orthogonal to h that passes through c'.

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. Let $\widehat{Q}_i = \{x_{i_1}, x_{i_2}\}$, $i = 1, \ldots, \lceil r/2 \rceil$, be a Tverberg partition of $x_1, \ldots, x_r$. (If r is odd, one of the sets 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 [4].

We claim that c is a Tverberg point for P of depth $\lceil r/2 \rceil$. Indeed, we have

$$c \in \mathrm{conv}(\widehat{Q}_i) = \mathrm{conv}(\{x_{i_1}, x_{i_2}\}) \subset \mathrm{conv}(P_{i_1} \cup P_{i_2}),$$

for $1 \le i \le \lceil r/2 \rceil$. Thus, if we set $Q_i := P_{i_1} \cup P_{i_2}$, then $Q_1, \ldots, Q_{\lceil r/2 \rceil}$ is a Tverberg partition for the point c. The total time to find c and the Q_i is $O(n)$, as claimed. See Figure 2 for a two-dimensional illustration of the lifting argument. □

The proof of Theorem 1.5 is now a direct consequence of Lemma 2.1.

PROOF OF THEOREM 1.5. If $d = 1$, we can immediately find a Tverberg point and a corresponding partition by finding the median c of P [4] and pairing each point to the left of the median with exactly one point to the right of the median.

If $d > 1$, we project the points onto the hyperplane defined by $x_d = 0$. This results in an n-point set $P' \subseteq \mathbb{R}^{d-1}$. We recursively find a Tverberg point and a Tverberg partition of depth $\lceil n/2^{d-1} \rceil$ for P', and then apply Lemma 2.1: in each step, the depth it at most halved. Thus, we end of with a point of depth at least $\lceil n/2^d \rceil$. We apply the pruning step described in the introduction to the resulting partition in each step, in order to ensure that the size of the resulting sets is $d+1$.

Then the running time for each level of the recursion is $d^{O(1)}n$, and there are d levels, which implies the result. □

In particular, we obtain a weak version of Tverberg's theorem with a very elementary proof.

COROLLARY 2.2 (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*

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

□

2.2 Improving the approximation factor

In order to improve the approximation factor, we will use an easy lemma to bootstrap the Tverberg depth.

LEMMA 2.3. *Suppose for any m-point set $Q \subseteq \mathbb{R}^d$ we can compute a point of Tverberg depth $\lceil m/\rho \rceil$ and a corresponding Tverberg partition in time $q(m,d)$. Let $P \subseteq \mathbb{R}^d$ with $|P| = n$, and let $c \in [2, n/\rho]$ be a constant. Then we can find $\alpha := \lceil \frac{n(1-1/c)}{\delta(d+1)} \rceil$ many Tverberg points of depth $\delta := \lceil n/c\rho \rceil$ for P and their corresponding partitions. These partitions have the additional property that each point of P appears in at most one partition. This takes total time*

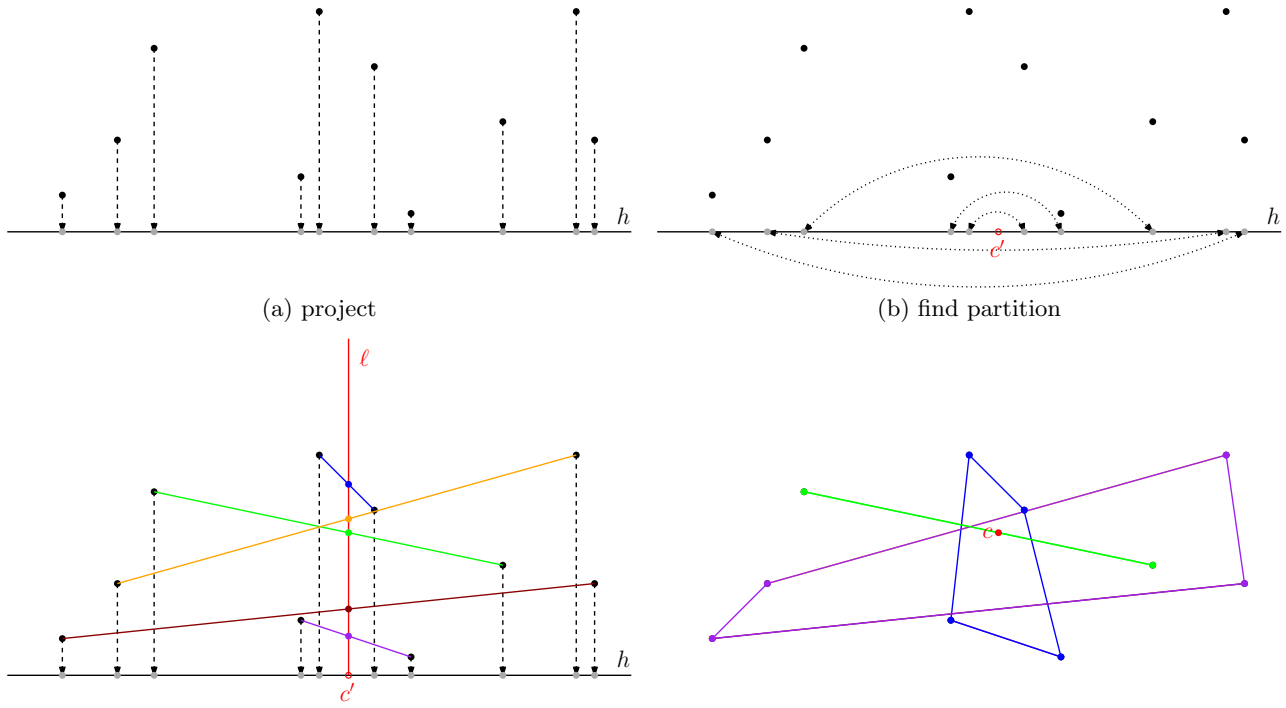
$$O\Big(\frac{(c-1)\rho}{d+1}(p(n,d) + q(n,d))\Big),$$

where $p(n,d)$ is the time for the pruning phase.

PROOF. Let $P_1 := P$. We take an arbitrary subset $P_1' \subseteq P_1$ with $\lceil n/c \rceil$ points and find a Tverberg point c_1 of depth $\lceil n/c\rho \rceil$ and a corresponding Tverberg partition $\mathcal{P}_1'$ for P_1'. Then we prune $\mathcal{P}_1'$ to get a Tverberg partition $\mathcal{P}_1$. Note that this takes time $p(n,d) + q(n,d)$ and that $Q_1 := \bigcup_{Z \in \mathcal{P}_1} Z$ contains at most $\lceil n/c\rho \rceil (d+1)$ points. Set $P_2 := P_1 \setminus Q_1$ and continue.

Each partition $\mathcal{P}_i$ partitions a set $Q_i \subseteq P$ such that the Q_i are pairwise disjoint. We can repeat this process until

$$n - i\delta(d+1) < \frac{n}{c},$$



(a) project (b) find partition

(c) intersect hulls of the sets with orthogonal complement (d) find median of intersections and combine

Figure 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.

which so solves to

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

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

For example, by Theorem 1.5 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 2.3 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 2.3, we will also need a lemma that states that sometimes by *combining* Tverberg points we can multiply their depth. This generalizes a similar lemma by Miller and Sheehy [10, Lemma 4.1].

Lemma 2.4. *Let P be a set of n points in $\mathbb{R}^d$, and let $P = \biguplus_{i=1}^{k} P_i$ 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 Tverberg partition $\mathcal{P}_i$. Let $C := \{c_i \mid 1 \leq i \leq k\}$ and c be a point of depth r' for C, with corresponding pruned Tverberg partition $\mathcal{C}$. Then c is a point of depth rr' for P. Furthermore, we can find a corresponding pruned Tverberg partition in time $d^{O(1)}n$.*

Proof. For $i = 1, \ldots, k$, 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. Clearly, 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}. We have

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

for $a = 1 \ldots r', b = 1 \ldots r$. Since certainly $|\mathcal{Z}| \leq n$, and since each Z_{ab} can be pruned in $d^{O(1)}$ time, the lemma follows. □

Combining Lemmas 2.3 and 2.4, we can now prove Theorem 1.6.

Proof of Theorem 1.6. If $n \leq 2^{d+1}$, we use the brute-force algorithm. This takes $f(2^{d+1})$ time.

Otherwise, we apply Lemma 2.3 (as in the remark following that lemma) to obtain a set C of

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

Tverberg points for P of depth $\lceil n/2^{d+1}\rceil$ with corresponding pruned partitions in linear time. We then use the brute-force algorithm to get a Tverberg point for C with depth $\lceil |C|/(d+1)\rceil$ with a corresponding partition, in time $f(|C|)$. Finally, we apply Lemma 2.4 to obtain a Tverberg point and corresponding partition in time $d^{O(1)}n$. The total running time is $f(2^d) + d^{O(1)}n$, and the resulting depth is

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

as desired. □

Alternatively, instead of the brute-force algorithm, we can use the algorithm by Miller and Sheehy to find a point of depth $\lceil n/4(d+1)^3\rceil$ in time $2^{O(d \log d)} + d^{O(1)}n$.

3. AN IMPROVED ALGORITHM

The algorithm in the previous section runs in linear time for any fixed dimension, but the constants are huge. Thus, finally we show how to improve our approach through an improved recursion and obtain an algorithm with running time $d^{O(\log d)}n$ while losing a depth factor of $1/2(d+1)$. First, however, we describe the more general version of Lemma 2.1 we promised above.

3.1 A more general version of the lifting argument

We present a more general version of the lifting argument in Lemma 2.1. For this we need some more notation. Let $P \subseteq \mathbb{R}^d$. Recall that 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 $\mathbb{R}^d$). A k-dimensional flat $F \subseteq \mathbb{R}^d$ is called 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 $\mathrm{conv}(P_i) \cap F \neq \emptyset$ for all $i = 1, \ldots, r$.

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 $\mathrm{pr}(P)$, 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. □

First of all, this shows how a good algorithm for any fixed dimension improves the general case:

COROLLARY 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 Tverberg 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 $n/\rho^{\lceil d/\delta\rceil}$ and a corresponding pruned partition in time $\lceil d/\delta\rceil f(n) + d^{O(1)}n$.

PROOF. We use induction on $k := \lceil d/\delta\rceil$ 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 Tverberg partition $P_1, \ldots, P_{\lceil n/\rho\rceil}$. For each i, we can 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 a $(d-\delta)$-dimensional point set. Since we have $\lceil (d-\delta)/\delta\rceil = k-1$, by induction we can find a Tverberg point c' for Q of depth $|Q|/\rho^{\lceil d/\delta\rceil - 1} = n/\rho^{\lceil d/\delta\rceil}$ and a corresponding pruned Tverberg 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, and 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. □

3.2 An improved algorithm

We will now show how to combine the above techniques for an algorithm with a better running time. The idea of the new algorithm is as follows: using Corollary 3.2, we reduce solving a d-dimensional instance to solving two instances of dimension $d/2$. This can be done recursively. Unfortunately, applying Corollary 3.2 reduces the depth of the partition. To fix this, we apply Lemmas 2.3, 2.4 and the Miller-Sheehy algorithm to increase the depth again. For clarity of presentation, we omit the roundings, which can be taken care of easily.

PROOF OF THEOREM 1.7. We prove the theorem by induction on d. As stated before, for $d = 1$ the claim is immediate, as in this case the problem reduces to a median computation.

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

Thus, by Corollary 3.2 (with $\delta = d/2$), there exists an algorithm that can compute a Tverberg point for P of depth $n/16(d/2+1)^6$ and a corresponding Tverberg partition in total time $2d^{\alpha \log(d/2)} + d^{O(1)}n$.

Now we apply Lemma 2.3. The lemma shows that we can compute $16(d/2+1)^6/(d+1)$ many points with depth

$n/32(d/2+1)^6$ and corresponding (disjoint) pruned partitions in time $d^{\alpha \log(d/2)+O(1)}n$.

Let C be the set of these Tverberg points. Applying the Miller-Sheehy algorithm, we can find a Tverberg point for C of depth $|C|/2(d+1)^2$ and a corresponding pruned Tverberg partition in time $|C|^{O(\log d)}$. Now, Lemma 2.4 shows that in additional $d^{O(1)}n$ time, we obtain a Tverberg point and a corresponding Tverberg partition for P of size

$$\frac{n}{2 \cdot 16(d/2+1)^6} \frac{16(d/2+1)^6}{2(d+1)^2(d+1)} = \frac{n}{4(d+1)^3},$$

as desired.

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

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

Since $|C| = d^{O(1)}$, we get

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

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

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

as claimed. This completes the proof. □

4. COMPARISON TO MILLER-SHEEHY

In the table below, we compare our algorithm in more detail to the Miller-Sheehy algorithm and its extensions. They give a generalization of their approach that shows that by computing higher order Tverberg points of depth r by brute-force, the running time can be improved for small d. This comes with the loss of factor r in the output. No exact values are given, but as far as we can tell, one can achieve a polynomial $O(f(d)n^2)$ running time for fixed d by setting the parameter $r = (d+1)$, while losing a factor of $(d+1)$ in the approximation. Further, even though it is not explicitly mentioned in the paper, we think that it is possible to also bootstrap their own algorithm (for a better running time in terms of d, while losing another factor of $(d+1)$ in the output). Table 4 shows a rough comparison of the different approaches. Again, f denotes the running time of the brute force algorithm.

Algorithm	**Running time**	**Depth**
Theorem 1.5	$O(dn)$	$n/2^d$
Miller-Sheehy	$n^{O(\log d)}$	$n/2(d+1)^2$
Theorem 1.6	$O\left(f(2^d) + d^{O(1)}n\right)$	$n/2(d+1)^2$
Miller-Sheehy generalized ($r = d+1$)	$O\left(f(d)n^2\right)$	$\approx n/2(d+1)^3$
Theorem 1.6 with M.-S.	$O\left(2^{O(d \log d)} + n\right)$	$n/4(d+1)^3$
Miller-Sheehy bootstrapped	$d^{O(\log d)}n^3$	$\approx n/2(d+1)^4$
Theorem 1.7	$d^{O(\log d)}n$	$n/4(d+1)^3$

5. CONCLUSION AND OUTLOOK

We have presented a very simple algorithm for finding an approximate Tverberg point, which runs in linear time for any fixed dimension. Using more sophisticated methods 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 guaranteed optimum.

Unfortunately, the resulting running time is still quasi-polynomial in d, and we still do not know whether there exists a polynomial algorithm (in n and d) for finding an approximate Tverberg point. However, we are hopeful that our techniques constitute a further step in this direction 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.

A promising approach, as also pointed out by one of the reviewers, would be to have a look at Sarkaria's proof of Tverberg's theorem using Colorful Carathéodory's theorem (see Matoušek [9, Chapter 8]). If one can compute a colorful simplex in truly polynomial time in any dimension, one can also compute a Tverberg point efficiently.

As mentioned before, the problem of deciding whether a given point has at least a certain depth is NP-complete. 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. [8]. However, this does not tell us anything about the actual problem of computing a point of depth $n/(d+1)$. As such a point is guaranteed to exist, it is not clear how to prove the problem to be hard using the "standard" NP-hardness reduction. Rather, we think that a hardness proof along the lines of complexity classes such as PPAD (see Papadimitriou [11]) should be pursued.

Acknowledgments. We would like to thank Nabil Mustafa for suggesting the problem to us. We also thank him and Don Sheehy for helpful discussions.

We would further like to thank the anonymous referees for their helpful and detailed comments.

6. REFERENCES

[1] T. M. Chan. An optimal randomized algorithm for maximum Tukey depth. In *Proc. 15th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 430–436, 2004.

[2] B. Chazelle. *The discrepancy method: randomness and complexity*. Cambridge University Press, Cambridge, 2000.

[3] K. L. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S.-H. Teng. Approximating center points with iterative Radon points. *Internat. J. Comput. Geom. Appl.*, 6(3):357–377, 1996.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, third edition, 2009.

[5] L. Danzer, B. Grünbaum, and V. Klee. Helly's theorem and its relatives. In *Proc. Sympos. Pure Math., Vol. VII*, pages 101–180. Amer. Math. Soc., Providence, R.I., 1963.

[6] H. Edelsbrunner. *Algorithms in combinatorial geometry*. Springer-Verlag, Berlin, 1987.

[7] S. Jadhav and A. Mukhopadhyay. Computing a centerpoint of a finite planar set of points in linear time. *Discrete Comput. Geom.*, 12(3):291–312, 1994.

[8] C. Knauer, H. R. Tiwary, and D. Werner. 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)*, volume 9, pages 649–660. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011.

[9] J. Matoušek. *Lectures on Discrete Geometry*. Springer, 2002.

[10] G. L. Miller and D. R. Sheehy. Approximate centerpoints with proofs. *Comput. Geom. Theory Appl.*, 43(8):647–654, 2010.

[11] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498 – 532, 1994.

[12] R. Rado. A theorem on general measure. *J. London Math. Soc.*, 21:291–300, 1946.

[13] S.-H. Teng. *Points, spheres, and separators: a unified geometric approach to graph partitioning*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1992.

[14] H. Tverberg. A generalization of Radon's theorem. *J. London Math. Soc.*, 41:123–128, 1966.