Approximating Tverberg Points in Linear Time for Any Fixed Dimension

Wolfgang Mulzer* Daniel Werner†‡

Abstract

Let P 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 $\operatorname{conv}(P_1), \ldots, \operatorname{conv}(P_r)$ 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 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 $n/4(d+1)^3$ 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 [8].

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 mesh generation) 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. A classic result in discrete geometry, the centerpoint theorem, shows that there exists a centerpoint for every point set [5, 9].

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'. If the dimension is not fixed, Teng shows that it is co-NP-hard to check whether a given point is a centerpoint [10].

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 centerpoints. 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 [8] 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. The resulting 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 $\log d$.

*Institut für Informatik, Freie Universität Berlin, Germany mulzer@inf.fu-berlin.de

†Institut für Informatik, Freie Universität Berlin, Germany dwerner@mi.fu-berlin.de

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

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 P that contains $n/4(d+1)^3$ 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 [7].

Theorem 1 (Radon's theorem) *For every set $P \subseteq \mathbb{R}^d$ with $d+2$ points there exists a partition (P_1, P_2) of P such that $conv(P_1) \cap conv(P_2) \neq \emptyset$.*

As mentioned above, Tverberg [11] generalized this theorem for larger point sets.

Theorem 2 (Tverberg's theorem) *Every 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 $\bigcap_{i=1}^{r} conv(P_i) \neq \emptyset$.*

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 $x \in \bigcap_{i=1}^{r} \mathrm{conv}(P_i)$. 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:

Theorem 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.*

By Carathéodory's theorem, in order to describe a Tverberg partition of depth r, we only need $r \cdot (d+1)$ points from P. This observation is also used by Miller and Sheehy [8]. 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*. A partition for which each set consists of $d+1$ points is called a *pruned partition*.

1.2 Our contribution

In Section 2, we present a simple lifting argument which leads to an easy Tverberg approximation algorithm.

Theorem 4 *Let P be a set of n points in $\mathbb{R}^d$ in general position. One can compute a Tverberg point of depth $n/2^d$ for P and the corresponding partition in time $O(n)$.*

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 applying the brute-force algorithm on small point sets, we get an even higher depth in linear time for any *fixed* dimension:

Theorem 5 *Let P be a set of n points in $\mathbb{R}^d$. Then one can compute a Tverberg point of depth $n/2(d+1)^2$ and a corresponding partition in time $f(2^d) + d^{O(1)}n$, where $f(m) = O(m!)$ is the time for computing a Tverberg point of depth $m/(d+1)$ for m points brute force.*

Finally, by combining our approach with that of Miller and Sheehy, we improve our algorithm to have a running time which is quasipolynomial in d:

Theorem 6 *Let P be a set of n points in $\mathbb{R}^d$. Then one can compute a Tverberg point of depth $n/4(d+1)^3$ and a corresponding partition in time $d^{O(\log d)} \cdot n$.*

2 A simple fixed parameter algorithm

A natural way to compute a Tverberg point for $P \subseteq \mathbb{R}^d$ 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 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, we then state the lemma in its full generality.

Lemma 7 *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 $O(n)$.*

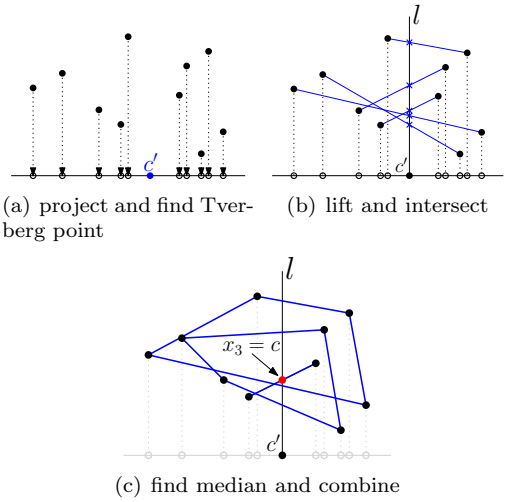
Proof. For every point $p \in P$, let $\mathrm{pr}(p)$ denote the projection of p onto h. Let $P_1, \ldots, P_r \subseteq P$ such that $\mathrm{pr}(P_1), \ldots, \mathrm{pr}(P_r)$ is a Tverberg partition for $\mathrm{pr}(P)$ with Tverberg point c'. Let ℓ be the line orthogonal to h that passes through 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 c. The total time to find c and the Q_i is $O(n)$, as claimed. See Figure 1 for a two-dimensional illustration of the lifting argument. □



(a) project and find Tverberg point (b) lift and intersect

(c) find median and combine

Figure 1: Illustrating the lifting lemma in the plane.

Proof. [of Theorem 4] We apply Lemma 7 recursively on the dimension, pruning the partitions after each lifting. This leads to an $O(n)$ running time. □

2.1 Improving the approximation factor

In order to improve the approximation factor, we will use an easy lemma to improve the Tverberg depth by collecting and combining several lower-depth points.

Lemma 8 *Let $P \subseteq \mathbb{R}^d$, $|P| = n$. If we can find a Tverberg point of depth n/ρ and the corresponding pruned partition in time $f(n,d)$, then we can find $\rho/(d+1)$ points of depth $n/2\rho$ in time $O\left(\frac{\rho}{(d+1)}\left(p(n,d) + f(n,d)\right)\right)$, where $p(n,d)$ is the time for the pruning phase.*

By Theorem 4 we can find a point of depth $n/2^d$ and a corresponding pruned partition in time $O(n)$. Thus, by applying Lemma 8 with $\rho = 2^d$, we can also find $2^d/(d+1)$ points of depth $n/2^{d+1}$ in linear time.

In order to make use of Lemma 8, we will 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 [8, Lemma 4.1].

Lemma 9 *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 $C := \{c_i \mid 1 \le i \le k\}$ and c be a point of depth r' for C, with corresponding pruned Tverberg partition $\mathcal{C}$. Then c is a point of depth $r \cdot r'$ for P. Furthermore, we can find a corresponding pruned Tverberg partition in time $d^{O(1)}n$.*

Theorem 5 then follows by combining Theorem 4 with Lemmas 8 and 9.

3 An improved algorithm

Finally we show how to improve our approach through a more sophisticated recursion and obtain an algorithm with running time $d^{O(\log d)} \cdot n$ for the price of losing a depth factor of $1/2(d+1)$. First, however, we describe the more general version of Lemma 7 we promised above.

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 $\mathbb{R}^d$. 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 10 *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 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.*

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

Corollary 11 *Let $\delta \ge 1$ be a fixed integer. Suppose we have an algorithm $\mathcal{A}$ with the following property: for every point set $Q \subseteq \mathbb{R}^\delta$, algorithm $\mathcal{A}$ constructs a Tverberg point of depth $|Q|/\rho$ 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 $n/\rho^{d/\delta}$ and a corresponding pruned partition in time $\lceil d/\delta \rceil f(n) + d^{O(1)}n$.

Proof. Induction on $k := \lceil d/\delta \rceil$ □

The idea of the new algorithm then is as follows: using Corollary 11, we reduce solving a d-dimensional instance to solving two instances of dimension $d/2$. This can be done recursively. Unfortunately, applying Corollary 11 reduces the depth of the partition. To fix this, we apply Lemmas 8, 9 and the Miller-Sheehy algorithm to increase the depth again. Details follow.

Proof. [of Theorem 6] We prove the theorem by induction on d. For $d = 1$ the claim is immediate: in this case the problem reduces to 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 11 (with $\delta = d/2$), there exists an algorithm that finds a Tverberg point for P of depth $n/16(d/2+1)^6$ and a corresponding Tverberg partition in time $2d^{\alpha \log(d/2)} + d^{O(1)}n$.

Now a more general version of Lemma 8 can be used to show that we can find $16(d/2+1)^6/(d+1)$ points of depth $n/32(d/2+1)^6$ and corresponding (disjoint) pruned partitions in time $d^{\alpha \log(d/2)+O(1)} \cdot n$.

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 9 shows that in additional $d^{O(1)}n$ time, we obtain a Tverberg point and a corresponding Tverberg partition for P of size

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

as desired.

It remains to analyze the running time. Adding up 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 $|C| = d^{O(1)}$, for α large enough $T(n,d) \leq d^{\alpha \log d}n = d^{O(\log d)}n$, as claimed.

□

4 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 combing our approach with known results, we managed to improve the running time to $d^{O(\log d)} \cdot n$, 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.

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 the anonymous reviewers for their helpful comments.

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

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

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

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

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