On β-Plurality Points in Spatial Voting Games

Citation for published version (APA):

Document license:
TAVERNE

DOI:
10.1145/3459097

Document status and date:
Published: 01/08/2021

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 06. Apr. 2022
On $\beta$-Plurality Points in Spatial Voting Games

BORIS ARONOV, Tandon School of Engineering, New York University, USA
MARK DE BERG, Department of Computing Science, TU Eindhoven, The Netherlands
JOACHIM GUDMUNDSSON and MICHAEL HORTON, School of Computer Science, University of Sydney, Australia

Let $V$ be a set of $n$ points in $\mathbb{R}^d$, called voters. A point $p \in \mathbb{R}^d$ is a plurality point for $V$ when the following holds: For every $q \in \mathbb{R}^d$, the number of voters closer to $p$ than to $q$ is at least the number of voters closer to $q$ than to $p$. Thus, in a vote where each $v \in V$ votes for the nearest proposal (and voters for which the proposals are at equal distance abstain), proposal $p$ will not lose against any alternative proposal $q$. For most voter sets, a plurality point does not exist. We therefore introduce the concept of $\beta$-plurality points, which are defined similarly to regular plurality points, except that the distance of each voter to $p$ (but not to $q$) is scaled by a factor $\beta$, for some constant $0 < \beta \leq 1$. We investigate the existence and computation of $\beta$-plurality points and obtain the following results.

- Define $\beta^*_d := \sup \{ \beta : \text{any finite multiset } V \text{ in } \mathbb{R}^d \text{ admits a } \beta \text{-plurality point} \}$. We prove that $\beta^*_2 = \sqrt{3}/2$, and that $1/\sqrt{d} \leq \beta^*_d \leq \sqrt{3}/2$ for all $d \geq 3$.
- Define $\beta(p, V) := \sup \{ \beta : p \text{ is a } \beta \text{-plurality point for } V \}$. Given a voter set $V$ in $\mathbb{R}^2$, we provide an algorithm that runs in $O(n \log n)$ time and computes a point $p$ such that $\beta(p, V) \geq \beta^*_2$. Moreover, for $d \geq 2$, we can compute a point $p$ with $\beta(p, V) \geq 1/\sqrt{d}$ in $O(n)$ time.
- Define $\beta(V) := \sup \{ \beta : V \text{ admits a } \beta \text{-plurality point} \}$. We present an algorithm that, given a voter set $V$ in $\mathbb{R}^d$, computes an $((1 - \epsilon) \cdot \beta(V))$-plurality point in time $O\left(\frac{n^2}{\epsilon^2} \cdot \log \frac{n}{\epsilon} \cdot \log^2 \frac{1}{\epsilon}\right)$.

CCS Concepts: • Theory of computation \rightarrow Computational geometry;

Additional Key Words and Phrases: Computational geometry, spatial voting theory, plurality point, computational social choice

Boris Aronov was partially supported by NSF Grants No. CCF-15-40656 and No. CCF-12-18791, and by Grant No. 2014/170 from the US-Israel Binational Science Foundation. Mark de Berg was supported by the Dutch Research Council (NWO) under project NETWORKS (Grant No. 024.002.003). Joachim Gudmundsson was supported under the Australian Research Council Discovery Projects funding scheme (Projects No. DP150101134 and No. DP180102870). Part of the work performed by Michael Horton was done while he was visiting NYU and was supported by NSF Grant No. CCF-12-18791.

Authors’ addresses: B. Aronov, Department of Computer Science & Engineering, Tandon School of Engineering, New York University, 370 Jay Street, Brooklyn, New York 11201, USA; email: boris.aronov@nyu.edu; M. de Berg, Department of Mathematics and Computer Science, PO Box 513, 5600 MB Eindhoven, the Netherlands; email: m.t.d.berg@tue.nl; J. Gudmundsson and M. Horton, School of Computer Science, University of Sydney, Building J12, 1 Cleveland Street, Darlington NSW 2008, Australia; emails: joachim.gudmundsson@sydney.edu.au, mhor9676@alumni.sydney.edu.au.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.
1549-6325/2021/07-ART24 $15.00
https://doi.org/10.1145/3459097

1 INTRODUCTION

Background. Voting theory is concerned with mechanisms to combine preferences of individual voters into a collective decision. A desirable property of such a collective decision is that it is stable, in the sense that no alternative is preferred by more voters. In spatial voting games [5, 10], this is formalized as follows; see Figure 1(i) for an example in a political context. The space of all possible decisions is modeled as $\mathbb{R}^d$ and every voter is represented by a point in $\mathbb{R}^d$, where the dimensions represent different aspects of the decision and the point representing a voter corresponds to the ideal decision for that voter. A voter $v$ now prefers a proposed decision $p \in \mathbb{R}^d$ over some alternative proposal $q \in \mathbb{R}^d$ when $v$ is closer to $p$ than to $q$. Thus, a point $p \in \mathbb{R}^d$ represents a stable decision for a given finite set $V$ of voters if, for any alternative $q \in \mathbb{R}^d$, we have $|\{v \in V : |vp| < |vq|\}| \geq |\{v \in V : |vp| < |vq|\}|$. Such a point $p$ is called a plurality point.\(^1\)

For $d = 1$, a plurality point always exists, since in $\mathbb{R}^1$ a median of $V$ is a plurality point. This is not true in higher dimensions, however. Define a median hyperplane for a set $V$ of voters to be a hyperplane $h$ such that both open half-spaces defined by $h$ contain fewer than $|V|/2$ voters. For $d \geq 2$ a plurality point in $\mathbb{R}^d$ exists if and only if all median hyperplanes for $V$ meet in a common point; see Figure 1(ii). This condition is known as generalized Plott symmetry conditions [12, 24]; see also the papers by Wu et al. [29] and de Berg et al. [4], who present algorithms to determine the existence of a plurality point for a given set of voters.

It is very unlikely that voters are distributed in such a way that all median hyperplanes have a common intersection. (Indeed, if this happens, then a slightest generic perturbation of a single voter destroys the existence of the plurality point.) It is unsatisfactory for the model to be unable to provide a solution in most cases, and so we may want to find a point that is close to being a plurality point. One way to formalize this is to consider the center of the yolk (or plurality ball) of $V$, where the yolk [14, 18, 22, 23] is the smallest ball intersecting every median hyperplane of $V$.

We introduce $\beta$-plurality points as an alternative way to relax the requirements for a plurality point, and study several combinatorial and algorithmic questions regarding $\beta$-plurality points.

$\beta$-Plurality points: definition and main questions. Let $V$ be a multiset\(^2\) of $n$ voters in $\mathbb{R}^d$ in arbitrary, possibly coinciding, positions. In the traditional setting a proposed point $p \in \mathbb{R}^d$ wins a voter $v \in V$ against an alternative $q$ if $|pv| < |qv|$. We relax this by fixing a parameter $\beta$ with $0 < \beta \leq 1$ and letting $p$ win $v$ against $q$ if $\beta \cdot |pv| < |qv|$. Thus, we give an advantage to the initial proposal $p$ by scaling distances to $p$ by a factor $\beta \leq 1$. We now define

$$V[p >_\beta q] := \{v \in V : \beta \cdot |pv| < |qv|\} \quad \text{and} \quad V[p <_\beta q] := \{v \in V : \beta \cdot |pv| > |qv|\}$$

to be the multisets of voters won by $p$ over $q$ and lost by $p$ against $q$, respectively. Finally, we say that a point $p \in \mathbb{R}^d$ is a $\beta$-plurality point for $V$ when

$$|V[p >_\beta q]| \geq |V[p <_\beta q]|,$$

for any point $q \in \mathbb{R}^d$.

\(^1\)One can also require $p$ to be strictly more popular than any alternative $q$. This is sometimes called a strong plurality point, in contrast to the weak plurality points that we consider.

\(^2\)Even though we allow $V$ to be a multiset, we sometimes refer to it as a “set” to ease the reading. When the fact that $V$ is a multiset requires special treatment, we explicitly address this.
Observe that \( \beta \)-plurality is monotone in the sense that if \( p \) is a \( \beta \)-plurality point then \( p \) is also a \( \beta' \)-plurality point for all \( \beta' < \beta \).

The spatial voting model was popularised by Black [5] and Down [10] in the 1950s. Stokes [27] criticized its simplicity and was the first to highlight the importance of taking non-spatial aspects into consideration. The reasoning is that voters may evaluate a candidate not only on their policies—their position in the policy space—but also take their so-called valence into account: charisma, competence, or other desirable qualities in the public’s mind [13]. A candidate can also increase her valence by a stronger party support [28] or campaign spending [19]. Several models have been proposed to bring the spatial model closer to a more realistic voting approach; see References [16, 17, 25] as examples. A common model is the multiplicative model, introduced by Hollard and Rossignol [20], which is closely related to the concept of a \( \beta \)-plurality point. The multiplicative model augments the existing spatial utility function by scaling the candidate’s valence by a multiplicative factor. Note that in the two-player game considered in this article the multiplicative model is the same as our \( \beta \)-plurality model. From a computational point of view very little is known about the multiplicative model. We are only aware of a result by Chung [8], who studied the problem of positioning a new candidate in an existing space of voters and candidates, so that the valence required to win at least a given number of voters is minimized.

One reason for introducing \( \beta \)-plurality was that a set \( V \) of voters in \( \mathbb{R}^d \), for \( d \geq 2 \), generally does not admit a plurality point. This immediately raises the question: Is it true that, for \( \beta \) small enough, any set \( V \) admits a \( \beta \)-plurality point? If so, then we want to know the largest \( \beta \) such that any voter set \( V \) admits a \( \beta \)-plurality point, that is, we wish to determine

\[
\beta^*_d := \sup \{ \beta : \text{any finite multiset } V \text{ in } \mathbb{R}^d \text{ admits a } \beta\text{-plurality point} \}.
\]

Note that \( \beta^*_1 = 1 \), since any set \( V \) in \( \mathbb{R}^1 \) admits a plurality point and 1-plurality is equivalent to the traditional notion of plurality.

After studying this combinatorial problem in Section 2, we turn our attention to the following algorithmic question: given a voter set \( V \), find a point \( p \) that is a \( \beta \)-plurality point for the largest possible value \( \beta \). In other words, if we define

\[
\beta(V) := \sup \{ \beta : V \text{ admits a } \beta\text{-plurality point} \}
\]
and
\[ \beta(p, V) := \sup(\beta : p \text{ is a } \beta\text{-plurality point for } V), \]
then we want to find a point \( p \) such that \( \beta(p, V) = \beta(V) \).

Results. In Section 2, we prove that \( \beta_d^* \leq \sqrt{3}/2 \) for all \( d \geq 2 \). To this end, we first show that \( \beta_d^* \) is non-increasing in \( d \), and then we exhibit a voter set \( V \in \mathbb{R}^2 \) such that \( \beta(V) \leq \sqrt{3}/2 \). To prove lower bounds on \( \beta_d^* \), we show that, for any given \( V \in \mathbb{R}^2 \), a point \( p \) exists such that \( \beta(p, V) \geq \sqrt{3}/2 \), thus proving that \( \beta_d^* = \sqrt{3}/2 \). Furthermore, we show how to construct such a point \( p \) in \( O(n \log n) \) time. Moreover, for \( d \geq 2 \), we prove the existence of—and show how to construct in \( O(n) \) time—a point \( p \) such that \( \beta(p, V) \geq 1/\sqrt{d} \), which means that \( \beta_d^* \geq 1/\sqrt{d} \).

In Section 3, we study the problem of computing, for a given voter set \( V \) of \( n \) points in \( \mathbb{R}^d \), a \( \beta \)-plurality point for the largest possible \( \beta \). (Here, we assume \( d \) to be a fixed constant.) While such a point can be found in polynomial time, the resulting running time is quite high. We therefore focus our attention on finding an approximately optimal point \( p \), that is, a point \( p \) such that \( \beta(p, V) \geq (1 - \varepsilon) \cdot \beta(V) \). We show that such a point can be computed in \( O\left(\frac{n^2}{\varepsilon^2d^2} \cdot \log \frac{n}{\varepsilon^2d^2} \cdot \log^2 \frac{1}{\varepsilon^2}\right) \) time.

Notation. We denote the open ball of radius \( r \) centered at a point \( q \in \mathbb{R}^d \) by \( B(q, r) \) and, for a point \( p \in \mathbb{R}^d \) and a voter \( v \), we define \( D_{p}(v) := B(v, \beta : |pv|) \). Observe that \( p \) wins \( v \) against a competitor \( q \) if and only if \( q \) is strictly inside \( D_{p}(v) \), while \( q \) wins \( v \) if and only if \( q \) is strictly outside \( D_{p}(v) \). Hence, \( V[p <_{D} q] = \{ v \in V : q \in D_{p}(v) \} \). We define \( D_{p}(p) := \{ D_{p}(p, v) : v \in V \} \)—here we assume \( V \) is clear from the context—and let \( A(D_{p}(p)) \) denote the arrangement induced by \( D_{p}(p) \). The competitor point \( q \) that wins the most voters against \( p \) will thus lie in the cell of \( A(D_{p}(p)) \) of the greatest depth or, more precisely, the cell contained in the maximum number of disks \( D_{p}(v) \).

2 BOUNDS ON \( \beta_{d}^* \)

In this section, we will prove bounds on \( \beta_{d}^* \), the supremum of all \( \beta \) such that any finite set \( V \subset \mathbb{R}^d \) admits a \( \beta \)-plurality point. We start with an observation that allows us to apply bounds on \( \beta_{d}^* \) to those on \( \beta_{d'}^* \) for \( d' > d \). Let \( \text{conv}(V) \) denote the convex hull of \( V \).

**Observation 2.1.** Let \( V \) be a finite multiset of voters in \( \mathbb{R}^d \).

(i) Suppose a point \( p \in \mathbb{R}^d \) is not a \( \beta \)-plurality point for \( V \). Then there is a point \( q \in \text{conv}(V) \) such that \( |V[p >_{\beta} q]| < |V[p <_{\beta} q]| \).

(ii) For any \( p' \notin \text{conv}(V) \), there is a point \( p \in \text{conv}(V) \) with \( \beta(p, V) > \beta(p', V) \).

(iii) For any \( d' > d \), we have \( \beta_{d'}^* \leq \beta_{d}^* \).

Proof. Note that for every point \( r \notin \text{conv}(V) \) there is a point \( r' \in \text{conv}(V) \) that lies strictly closer to all voters in \( V \), namely, the point \( r' \in \partial \text{conv}(V) \) closest to \( r \). This immediately implies part (i): if \( p \) is beaten by some point \( q \notin \text{conv}(V) \) then \( p \) is certainly beaten by a point \( q' \in \text{conv}(V) \) that lies strictly closer to all voters in \( V \) than \( q \). It also immediately implies part (ii), because if a point \( p \) lies strictly closer to all voters in \( V \) than a point \( p' \), then \( \beta(p, V) > \beta(p', V) \).

To prove part (iii), let \( V \in \mathbb{R}^d \) be a voter set such that \( \beta(V) = \beta_{d}^* \). Now embed \( V \) into \( \mathbb{R}^{d'} \), say in the flat \( x_{d+1} = \cdots = x_d = 0 \), obtaining a set \( V' \). Then \( \beta(V') = \beta(V) \) by parts (i) and (ii). Hence, \( \beta_{d'}^* \leq \beta(V') = \beta(V) = \beta_{d}^* \).

\[ \square \]

\[ ^{\text{3}} \text{Very recently, Filtscher and Filtscher} \cite{FiltscherFiltscher21} \text{ improved these results for } d \geq 4 \text{ by proving that } \beta_{d}^* \geq \frac{1}{2} \sqrt{\frac{1}{2} + \frac{1}{2} \sqrt{3} - \frac{1}{2} \sqrt{4 \sqrt{3} - 3} \approx 0.557 \text{ for any } d \geq 4.} \]
We can now prove an upper bound on $\beta_d^*$.  

**Lemma 2.2.** $\beta_d^* \leq \sqrt{3}/2$, for $d \geq 2$.  

**Proof.** By Observation 2.1(iii), it suffices to prove the lemma for $d = 2$. To this end let $V = \{v_1, v_2, v_3\}$ consist of three voters that form an equilateral triangle $\Delta$ of side length 2 in $\mathbb{R}^2$; see Figure 2(i).  

Let $p$ denote the center of $\Delta$. We will first argue that $\beta(p, V) = \sqrt{3}/2$. Note that $|pv_i| = 2/\sqrt{3}$ for all three voters $v_i$. Hence, for $\beta = \sqrt{3}/2$, the open balls $D_\beta(v_i, p)$ are pairwise disjoint and touching at the mid-points of the edges of $\Delta$. Therefore any competitor $q$ either wins one voter and loses the remaining two, or wins no voter and loses at least one. The former happens when $q$ lies inside one of the three balls $D_\beta(v_i, p)$; the latter happens when $q$ does not lie inside any of the balls, because in that case $q$ can be on the boundary of at most two of the balls. Thus, for $\beta = \sqrt{3}/2$, the point $p$ always wins more voters than $q$ does. However, for $\beta > \sqrt{3}/2$, any two balls $D_\beta(v_i, p), D_\beta(v_j, p)$ intersect and so a point $q$ located in such a pairwise intersection wins two voters and beats $p$. We conclude that $\beta(p, V) = \sqrt{3}/2$, as claimed. 

The lemma now follows if we can show that $\beta(p', V) \leq \sqrt{3}/2$ for any $p' \neq p$. Let $\text{Vor}(V)$ be the Voronoi diagram of $V$, and let $\mathcal{V}(v_i)$ be the closed Voronoi cell of $v_i$, as shown in Figure 2(ii). Assume without loss of generality that $p'$ lies in $\mathcal{V}(v_3)$. Let $E$ be the ellipse with foci $v_1$ and $v_2$ that passes through $p$. Thus, 

$$E := \left\{ z \in \mathbb{R}^2 : |zv_1| + |zv_2| = 4/\sqrt{3} \right\}. $$

Note that $E$ is tangent to $\mathcal{V}(v_3)$ at the point $p$. Hence, any point $p' \neq p$ in $\mathcal{V}(v_3)$ has $|p'v_1| + |p'v_2| > 4/\sqrt{3}$. This implies that for $\beta \geq \sqrt{3}/2$, we have $\beta \cdot |p'v_1| + \beta \cdot |p'v_2| > 2$, and so the disks $D_\beta(p', v_1)$ and $D_\beta(p', v_2)$ intersect. It follows that for $\beta \geq \sqrt{3}/2$ there is a competitor $q$ that wins two voters against $p'$, which implies $\beta(p', V)$ cannot be larger than $\sqrt{3}/2$ and thus finishes the proof of the lemma. 

We now prove lower bounds on $\beta_d^*$. We first prove that $\beta_d^* \geq 1/\sqrt{d}$ for any $d \geq 2$, and then we improve the lower bound to $\sqrt{3}/2$ for $d = 2$. The latter bound is tight by Lemma 2.2.  

Let $V$ be a finite multiset of $n$ voters in $\mathbb{R}^d$. We call a hyperplane $h$ balanced with respect to $V$, if both open half-spaces defined by $h$ contain at most $n/2$ voters from $V$. Note the difference with median hyperplanes, which are required to have fewer than $n/2$ voters in both open half-spaces. Clearly, for any $1 \leq i \leq d$ there is a balanced hyperplane orthogonal to the $x_i$-axis, namely, the hyperplane $x_i = m_i$, where $m_i$ is a median in the multiset of all $x_i$-coordinates of the voters in $V$. (In fact, for any direction $\vec{u}$ there is a balanced hyperplane orthogonal to $\vec{u}$.)
LEMMA 2.3. Let $d \geq 2$. For any finite multiset $V$ of voters in $\mathbb{R}^d$ there exists a point $p \in \mathbb{R}^d$ such that $\beta(p, V) = 1/\sqrt{d}$. Moreover, such a point $p$ can be computed in $O(n)$ time.

Proof. Let $\mathcal{H} := \{h_1, \ldots, h_d\}$ be a set of balanced hyperplanes with respect to $V$ such that $h_i$ is orthogonal to the $x_i$-axis, and assume without loss of generality that $h_i$ is the hyperplane $x_i = 0$. We will prove that the point $p$ located at the origin is a $\beta$-plurality point for $V$ for any $\beta < 1/\sqrt{d}$, thus showing that $\beta(p, V) \geq 1/\sqrt{d}$.

Let $q = (q_1, \ldots, q_d)$ be any competitor of $p$. We can assume without loss of generality that $\max_{1 \leq i \leq d} |q_i| = q_d > 0$. Thus, $q$ lies in the closed cone $C_d^+$ defined as

$$C_d^+ := \left\{ (x_1, \ldots, x_d) \in \mathbb{R}^d : x_d \geq |x_j| \text{ for all } j \neq d \right\}.$$

Note that $C_d^+$ is bounded by portions of the $2(d - 1)$ hyperplanes $x_d = \pm x_j$ with $j \neq d$; see Figure 3.

Because $h_d : x_d = 0$ is a balanced hyperplane, the open halfspace $h_d^+ : x_d > 0$ contains at most $n/2$ voters, which implies that the closed halfspace $\mathrm{cl}(h_d^-) : x_d \leq 0$ contains at least $n/2$ voters. Hence, it suffices to argue that for any $\beta < 1/\sqrt{d}$ the point $p$ wins all the voters in $\mathrm{cl}(h_d^-)$ against $q$.

Claim. For any voter $v \in \mathrm{cl}(h_d^-)$ with $|pv| > |qv|$, $\sin(\angle qpv) \geq 1/\sqrt{d}$ with equality if and only if $q$ lies on an edge of $C_d^+$ and $v$ lies on the orthogonal projection of this edge onto $h_d$.

Proof. Consider a voter $v \in \mathrm{cl}(h_d^-)$ with $|pv| > |qv|$. If $v$ lies strictly below $h_d$, then there is a point $v' \in h_d$ with $\angle qpv' < \angle qpv$, namely, the orthogonal projection of $v$ onto $h_d$. Since $|pv| > |qv|$, we have $\angle qpv < \pi/2$, and so $\angle qpv' < \angle qpv$ implies that $\sin(\angle qpv') < \sin(\angle qpv)$. Hence, it suffices to prove the claim for $v \in h_d$.

First, we prove that $\sin(\angle qpu) = 1/\sqrt{d}$ if $q$ lies on an edge $e$ of $C_d^+$ and $v$ lies on the orthogonal projection $\overline{v}$ of $e$ onto $h_d$. Assume without loss of generality that $e$ is the edge of $C_d^+$ defined by the intersection of the $d - 1$ hyperplanes $x_d = x_j$, so that $q_1 = \cdots = q_{d-1} = q_d$. Since $\angle qpu$ is the same for any $v \in \overline{e}$, we may assume that $v$ is the orthogonal projection of $q$ to $h_d$, which means $|qv| = q_d$. We then have

$$\sin(\angle qpu) = \frac{|qv|}{|pq|} = \frac{q_d}{\sqrt{q_1^2 + \cdots + q_d^2}} = \frac{1}{\sqrt{d}}.$$
Now assume the condition for equality does not hold. Let ρ be the ray starting at p and containing q, and let ρ̂ be its orthogonal projection onto hd. We have two cases: v ∈ ρ̂ but q is not contained in an edge of C^+_d, or v /∈ ρ̂.

In the former case, we may, as before, assume that v is the projection of q onto hd. Since q ∈ C^+_d, we have q_d ≥ |q_j| for all j. Moreover, since q does not lie on an edge of C^+_d, we have q_d > |q_j| for at least one j. Thus, |pq| = \sqrt{q_1^2 + \ldots + q_d^2} > \sqrt{d} \cdot |qv|, and \sin(∠qpv) = |qv|/|pq| > 1/\sqrt{d}.

In the latter case, let ℓ be the line containing p and v, and let v be the point on ℓ closest to q. Then |qv| ≥ |qv'| > |q̅q|, where q̅ is the projection of q onto hd. Since we assumed |pv| > |qv| we have ∠qpv < π/2, which implies that v and v lie to the same side of p. Hence, ∠qpv = ∠qpv', and we have

\[ \sin(∠qpv) = \frac{|qv'|}{|pq|} > \frac{|q̅q|}{|pq|} ≥ \frac{1}{\sqrt{d}}. \]

Observe that p trivially wins every voter with |pv| ≤ |qv|. For the voters v ∈ cl(h^-_d) with |pv| > |qv| we can use the Law of Sines and the claim above to derive that

\[ \beta \cdot |pv| < \frac{1}{\sqrt{d}} \cdot |pv| = \frac{1}{\sqrt{d}} \cdot \frac{|qv| \cdot \sin(∠pqv)}{\sin(∠qpv)} ≤ \frac{|qv| \cdot \sin(∠pqv)}{1} ≤ |qv|. \]

Hence, p wins every voter in cl(h^-_d). This proves the first part of the lemma, since cl(h^-_d) contains at least n/2 voters, as already remarked.

Computing the point p is trivial once we have the balanced hyperplanes hd, which can be found in O(n) time by computing a median x_i-coordinate for each 1 ≤ i ≤ d.

A tight bound in the plane. In R^2, we can improve the above bound: for any voter set V in the plane, we can find a point p with \( \beta(p, V) \geq \sqrt{3}/2 \). By Lemma 2.2, this bound is tight. The improvement is based on the following lemma.

**Lemma 2.4.** Let V be a multiset of n voters in R^2, let ℓ_1, ℓ_2, ℓ_3 be a triple of concurrent balanced lines such that the smaller angle between any two of them is π/3, and let p be the common intersection of ℓ_1, ℓ_2, ℓ_3. Then \( \beta(p, V) \geq \sqrt{3}/2 \).
Fig. 5. (i) The balanced line \( \mu(\theta) \). (ii) If \( p_{23} \) is to the left of the directed line \( \ell_1(0) \), then \( p_{13}(0) \) is to the right of \( \ell_2(0) \).

Proof. Let \( q \) be a competitor of \( p \). The three lines \( \ell_1, \ell_2, \ell_3 \) partition the plane into six equal-sized sectors, which we number \( S_1 \) through \( S_6 \) in a clockwise fashion, so that \( q \) lies in the closure of \( S_1 \); see Figure 4. Let \( H \) be the closure of \( S_3 \cup S_4 \cup S_5 \). It is a closed halfspace bounded by a balanced line, so it contains at least half the voters.

Using an analysis similar to that in the proof of Lemma 2.3, we can show that \( p \) does not lose any voter \( v \in H \). Indeed, using the Law of Sines, we obtain

\[
\frac{\sqrt{3}}{2} \cdot |pv| = \frac{\sqrt{3}}{2} \cdot \frac{\sin \angle qpv}{\sin \angle qpv} \cdot |qv| \leq |qv|, \quad \text{since} \quad \angle qpv \geq \pi/3,
\]

which shows that \( p \) is a \( \beta \)-plurality point for any \( \beta < \sqrt{3}/2 \). Hence, \( \beta(p, V) \geq \sqrt{3}/2 \). \( \square \)

The main question is whether a triple of concurrent lines as in Lemma 2.4 always exists. The next lemma shows that this is indeed the case. The lemma—in fact, a stronger version, stating that any two opposite cones defined by the three concurrent lines contain the same number of points—has been proved for even \( n \) by Dumitrescu et al. \[11\]. Our proof of Lemma 2.5 is similar to their proof. We give it because we also need it for odd \( n \), and because we will need an understanding of the proof to describe our algorithm for computing the concurrent triple in the lemma. Our algorithm will run in \( O(n \log n) \) time, a significant improvement over the \( O(n^{1/3} \log^{1+\epsilon} n) \) running time obtained (for the case of even \( n \)) by Dumitrescu et al. \[11\].

Lemma 2.5. For any multiset \( V \) of \( n \) voters in \( \mathbb{R}^2 \), there exists a triple of concurrent balanced lines \( (\ell_1, \ell_2, \ell_3) \) such that the smaller angle between any two of them is \( \pi/4 \).

Proof. Define the orientation of a line to be the counterclockwise angle it makes with the positive \( y \)-axis. Recall that for any given orientation \( \theta \) there exists at least one balanced line with orientation \( \theta \). When \( n \) is odd this line is unique: it passes through the median of the voter set \( V \) when \( V \) is projected orthogonally onto a line orthogonal to the lines of orientation \( \theta \). In the rest of the proof it will be convenient to have a unique balanced line for any orientation \( \theta \). To achieve this when \( n \) is even, we simply delete an arbitrary voter from \( V \). (If there are other voters at the same location, then these voters are not deleted.) This is allowed because when \( |V| \) is even, a balanced line for \( V \setminus \{v\} \) is also a balanced line for \( V \).

Now let \( \mu \) be the function that maps an angle value \( \theta \) to the unique balanced line \( \mu(\theta) \); see Figure 5(i). Note that \( \mu \) is continuous for \( 0 \leq \theta < \pi \). Let \( \ell_1(\theta) := \mu(\theta) \), and \( \ell_2(\theta) := \mu(\theta + \pi/4) \), and \( \ell_3(\theta) := \mu(\theta + \pi/2) \). For \( i \neq j \), let \( p_{ij}(\theta) := \ell_i(\theta) \cap \ell_j(\theta) \) be the intersection point between \( \ell_i(\theta) \) and \( \ell_j(\theta) \). If \( p_{23}(0) \in \ell_1(0) \), then the lines \( \ell_1(0), \ell_2(0), \ell_3(0) \) are concurrent and we are done. Otherwise, consider the situation at \( \theta = 0 \) and imagine \( \ell_1(0) \) and \( \ell_2(0) \) to be directed in the positive \( y \)-direction, as in Figure 5(ii). Clearly, if \( p_{23}(0) \) is to the left of the directed line \( \ell_1(0) \), then \( p_{13}(0) \) is to the right of the directed line \( \ell_2(0) \), and vice versa. Now increase \( \theta \) from 0 to \( \pi/3 \), and note that \( \ell_1(\pi/3) = \ell_2(0) \) and \( p_{23}(\pi/3) = p_{13}(0) \). Hence, \( p_{23}(\theta) \) lies to a different side of the directed
line $\ell_1(\theta)$ for $\theta = 0$ than it does for $\theta = \pi/3$. Since both $\ell_1(\theta)$ and $p_{23}(\theta)$ vary continuously with $\theta$, this implies that, for some $\theta \in (0, \pi/3)$, the point $p_{23}(\theta)$ crosses the line $\ell_1(\theta)$, and so the lines $\ell_1(\theta), \ell_2(\theta), \ell_3(\theta)$ are concurrent.

The previous two lemmas show that any voter set $V$ in $\mathbb{R}^2$ admits a point $p$ such that $\beta(p, V) \geq \sqrt{3}/2$. We now show that we can compute such a point in $O(n \log n)$ time, namely, we show how to compute a triple as in Lemma 2.5 in $O(n \log n)$ time. We follow the definitions and notation from the proof of that lemma. We will assume that $n$ is odd, which, as argued, is without loss of generality.

To find a concurrent triple of balanced lines, we first compute the lines $\ell_1(0), \ell_2(0), \ell_3(0)$ in $O(n)$ time. If they are concurrent, then we are done. Otherwise, there is a $\bar{\theta} \in (0, \pi/3)$ such that $\ell_1(\bar{\theta}), \ell_2(\bar{\theta}), \ell_3(\bar{\theta})$ are concurrent. To find this value $\bar{\theta}$, we dualize the voter set $V$, using the standard duality transform that maps a point $(a, b)$ to the non-vertical line $y = ax - b$ and vice versa. Let $v^*$ denote the dual line of the voter $v$, and let $V^* := \{v^* : v \in V\}$. Note that for $\bar{\theta} \in (0, \pi/3)$ the lines $\ell_1(\bar{\theta}), \ell_2(\bar{\theta}), \ell_3(\bar{\theta})$ are all non-vertical; therefore, their duals $\ell_1^*(\bar{\theta}), \ell_2^*(\bar{\theta}), \ell_3^*(\bar{\theta})$ are well-defined.

Consider the arrangement $\mathcal{A}(V^*)$ defined by the duals of the voters. For $\theta \neq 0$, define slope($\theta$) to be the slope of the lines with orientation $\theta$. Then $\mu^*(\theta)$, the dual of $\mu(\theta)$, is the intersection point of the vertical line $x = \text{slope}(\theta)$ with $L_{\text{med}}$, the median line in $\mathcal{A}(V^*)$. (The median level of $\mathcal{A}(V^*)$ is the set of points $q$ such that there are fewer than $n/2$ lines below $q$ and fewer than $n/2$ lines above $q$; this is well defined, since we assume $n$ is odd. The median level forms an $x$-monotone polygonal curve along the edges of $\mathcal{A}(V^*)$.) Observe that the duals $\ell_1^*(\theta), \ell_2^*(\theta), \ell_3^*(\theta)$ all lie on $L_{\text{med}}$. For $\theta \in (0, \pi/3)$, the $x$-coordinate of $\ell_1^*(\theta)$ lies in $(-\infty, -1/\sqrt{3})$, the $x$-coordinate of $\ell_2^*(\theta)$ lies in $(-1/\sqrt{3}, 1/\sqrt{3})$, and the $x$-coordinate of $\ell_3^*(\theta)$ lies in $(1/\sqrt{3}, \infty)$. We split $L_{\text{med}}$ into three pieces corresponding to these ranges of the $x$-coordinate. Let $E_1, E_2, E_3$ denote the sets of edges forming the parts of $L_{\text{med}}$ in the first, second, and third range, respectively, where edges crossing the vertical lines $x = -1/\sqrt{3}$ and $x = 1/\sqrt{3}$ are split; see Figure 6. Thus, for any $\theta \in (0, \pi/3)$ and for $i \in \{1, 2, 3\}$, the point $\ell_i^*(\theta)$ lies on an edge in $E_i$.

Recall that we want to find a value $\bar{\theta} \in (0, \pi/3)$ such that $\ell_1(\bar{\theta}), \ell_2(\bar{\theta}), \ell_3(\bar{\theta})$ are concurrent. For $-\infty < x < 1/\sqrt{3}$, let $\theta_x$ be such that slope($\theta_x$) = $x$, and for $i \in \{1, 2, 3\}$ define $p_i(x) := \ell_i^*(\theta_x)$. We are thus looking for a value $\bar{x} \in (-\infty, 1/\sqrt{3})$ such that the three points $p_1(\bar{x}), p_2(\bar{x}), p_3(\bar{x})$ are collinear.

One way to find $\bar{x}$ would be to first explicitly compute $L_{\text{med}}$; then we can increase $x$, starting at $x = -\infty$, and see how the points $p_i(x)$ move over $E_i$, until we reach a value $\bar{x}$ such that $p_1(\bar{x}), p_2(\bar{x}), p_3(\bar{x})$ are collinear. Since the best-known bounds on the complexity of the median level is $O(n^{113})$ [9], we will proceed differently, as follows.

1. Use the recursive algorithm described below to find an interval $I \subseteq (-\infty, 1/\sqrt{3})$ containing a value $\bar{x}$ with the desired property—namely, that the points $p_1(\bar{x}), p_2(\bar{x}), p_3(\bar{x})$ are collinear—and such that, for any $i \in \{1, 2, 3\}$, the point $p_i(x)$ lies on the same edge of $E_i$ for all $x \in I$.  

![Diagram of edge sets](https://example.com/diagram.png)

Fig. 6. The edge sets $E_1, E_2, E_3$ of $L_{\text{med}}$, the median level in $\mathcal{A}(V^*)$. 

(2) Find a value \( \bar{x} \in \bar{I} \) such that \( p_1(\bar{x}), p_2(\bar{x}), \) and \( p_3(\bar{x}) \) are collinear. Since for \( i = 1, 2, 3 \) each \( p_i(x) \) lies on a fixed edge \( e_i \) of \( L_{med} \) for all \( x \in \bar{I} \) after Step 1, this can be done in \( O(1) \) time. Indeed, if we go back to primal space, we are given three (not necessarily distinct) voters \( v_1, v_2, v_3 \) (namely, the primals of the lines containing \( e_1, e_2, \) and \( e_3 \) and a range \( (\theta, \theta') \) of angles (corresponding to the \( x\)-range \( \bar{I} \)), such that the line \( \ell_i(\bar{\theta}) \) passes through \( v_i \) for any \( \bar{\theta} \in (\theta, \theta') \). We then only have to compute an orientation \( \bar{\theta} \in (\theta, \theta') \) such that the lines \( \ell_i(\bar{\theta}) \) meet in a common point—such an orientation \( \bar{\theta} \) is guaranteed to exist by our construction of \( \bar{I} \).

We now explain the recursive algorithm used in Step 1. In a generic call, we are given three trapezoids \( \Delta_1, \Delta_2, \Delta_3 \) that are each bounded by two non-vertical edges and two vertical edges (one of which may degenerate into a point). Let \( I_i \) be the \( x\)-range of \( \Delta_i \), for \( i \in \{1, 2, 3\} \); note that this is well-defined, since \( \Delta_1 \) is a trapezoid delimited by two vertical edges. The trapezoids \( \Delta_1, \Delta_2, \Delta_3 \) will have the following properties.

(P1) Trapezoid \( \Delta_1 \) lies inside the vertical slab \( (-\infty, -1/\sqrt{3}) \times (-\infty, \infty) \), trapezoid \( \Delta_2 \) lies inside the vertical slab \( (-1/\sqrt{3}, 1/\sqrt{3}) \times (-\infty, \infty) \), and trapezoid \( \Delta_3 \) lies inside the vertical slab \( (1/\sqrt{3}, \infty) \times (-\infty, \infty) \). Moreover, the \( x\)-ranges \( I_1, I_2, I_3 \) correspond to each other in the following sense. Recall that an \( x\)-range \( I \subset (-\infty, -1/\sqrt{3}) \) in the dual plane corresponds to an angular interval \( (\phi, \phi') \) in the primal plane. We denote by \( I \oplus \theta \) the \( x\)-range corresponding to the angular interval \( (\phi + \theta, \phi' + \theta) \). The \( x\)-ranges \( I_1, I_2, I_3 \) will be such that \( I_2 = I_1 \oplus \frac{\pi}{3} \) and \( I_3 = I_1 \oplus \frac{2\pi}{3} \).

(P2) For any \( i \in \{1, 2, 3\} \), the part of the median level \( L_{med} \) inside the vertical slab \( I_i \times (-\infty, \infty) \) lies entirely inside \( \Delta_i \). Together with the property (P1) this implies that for any \( x \in I_i \), we have that \( p_1(x) \in \Delta_i \), for \( i \in \{1, 2, 3\} \).

(P3) Let \( x_{\text{left}} \) and \( x_{\text{right}} \) be such that \( I_1 = (x_{\text{left}}, x_{\text{right}}) \). Then, we have: If \( p_1(x_{\text{left}}) \) lies above the line through \( p_2(x_{\text{left}}) \) and \( p_3(x_{\text{left}}) \), then \( p_1(x_{\text{right}}) \) lies below the line through \( p_2(x_{\text{right}}) \) and \( p_3(x_{\text{right}}) \), and vice versa. This guarantees that there exists a value \( \bar{x} \in I_1 \) with the desired property and, hence, that \( I_1 \) contains the interval \( \bar{I} \) we are looking for.

In a recursive call, we are also given for each \( \Delta_i \) the sets \( V_i^* \subseteq V^* \) of lines intersecting the interior of \( \Delta_i \), as well as \( n_i^- \), the number of lines from \( V^* \) passing completely below \( \Delta_i \).

Initially, \( \Delta_1 \) is the unbounded trapezoid\(^4\) \( (-\infty, -1/\sqrt{3}) \times (-\infty, \infty) \). Similarly, we initially have \( \Delta_2 = (-1/\sqrt{3}, 1/\sqrt{3}) \times (-\infty, \infty) \) and \( \Delta_3 = (1/\sqrt{3}, \infty) \times (-\infty, \infty) \). Furthermore, \( V_1^* = V^* \) and \( n_1^- = 0 \) for \( i = 1, 2, 3 \).

The recursion ends when the interior of each \( \Delta_i \) is intersected by a single edge of \( L_{med} \); we then have \( \bar{I} := I_1 \). The recursive call for a given triple \( \Delta_1, \Delta_2, \Delta_3 \) starts by shrinking \( \Delta_1 \) to a trapezoid \( \Delta'_1 \)—thus zooming in on the value \( \bar{x} \)—as follows. (We assume \( |V_1^*| > 1 \), otherwise the shrinking of \( \Delta_1 \) can be skipped.)

(i) Set \( r := 2 \) and construct a \((1/r)\)-cutting for the lines in \( V_1^* \), clipped to within \( \Delta_1 \). In other words, construct a partition \( \Xi \) of \( \Delta_1 \) into \( O(r^2) = O(1) \) smaller trapezoids—see

\(^4\)Since \( x_{\text{left}} = -\infty \) for this initial trapezoid \( \Delta_1 \), the points \( p_1(x_{\text{left}}) \) are not well defined. Recall, however, that in the primal plane the point on \( L_{med} \) "at \( x = -\infty \)" corresponds to the vertical balanced line \( \ell_1(0) \). Hence, we can derive the relative position of \( p_1(-\infty) \) with respect to the line through \( p_2(-\infty) \) and \( p_3(-\infty) \), from the relative position of the intersection point \( \ell_2(0) \cap \ell_3(0) \) with respect to \( \ell_1(0) \).
Figure 7—such that the interior of each trapezoid $\tau \in \Xi$ is intersected by at most $n_1/2$ lines from $V_1^*$, where $n_1 := |V_1^*|$. Computing $\Xi$ can be done in $O(n_1)$ time [6].

(ii) Compute the intersections of $L_{\text{med}}$ with the edges of each trapezoid $\tau \in \Xi$ in $O(n_1 \log n_1)$ time, as follows.

- Consider a non-vertical edge $e$ of $\tau$. First, compute the level of $p_{\text{left}}$, the left endpoint of $e$. This can be done by counting the number of lines from $V_1^*$ below $p_{\text{left}}$ and adding $n_1 - 1$ to this number. Next, intersect $e$ with all lines of $V_1^*$ and sort the intersections along $e$, distinguishing lines that cross $e$ “from above” and “from below.” Finally, walk along $e$, starting from $p_{\text{left}}$ towards the right, increasing and decreasing the level according to the type of the intersection we encounter. All intersection points that lie on $L_{\text{med}}$ can thus be reported. (For simplicity, we ignore the case where $e$ partially or fully overlaps $L_{\text{med}}$. This can either be handled by a simple modification of the procedure, or we can avoid the situation altogether by modifying the cutting such that no edge $e$ of the cutting is contained in an input line.)

- For a vertical edge $e$ of $\tau$, we proceed similarly: first compute the level of the lower endpoint of $e$, and then walk upward along $e$ until we reach an intersection point at the median level, or the upper endpoint of $e$.

(iii) The previous step gives us all intersection points of $L_{\text{med}}$ with the edges of trapezoids in $\Xi$. Let $X = \{\xi_1, \ldots, \xi_{|X|}\}$ be the sorted set of all $x$-coordinates of these intersection points, as illustrated in Figure 7. Note that $|X| = O(n_1)$. We perform a binary search on $X$ to find two consecutive $x$-coordinates, $\xi_i$ and $\xi_{i+1}$, such that the interval $(\xi_i, \xi_{i+1})$ contains a value $\bar{x}$ with the desired property. Each step in the binary search can be done in $O(n_1 + n_2 + n_3)$ time, as follows.

Assume without loss of generality that $p_1(x_{\text{left}})$ lies above the line through $p_2(x_{\text{left}})$ and $p_3(x_{\text{left}})$, and that $p_1(x_{\text{right}})$ lies below the line through $p_2(x_{\text{right}})$ and $p_3(x_{\text{right}})$. Suppose that during the binary search we arrive at some $\xi_j \in X$, and we want to decide if we want to proceed to the left or to the right of $\xi_j$. To do so, we first compute the points $p_1(\xi_j)$, $p_2(\xi_j)$, and $p_3(\xi_j)$. Point $p_1(\xi_j)$ can be computed in $O(n_1)$ time, as follows: first compute all intersections of the vertical line $x = \xi_j$ with the lines in $V_1^*$, and then find the intersection point whose $y$-coordinate has the appropriate rank, taking into account that there are $n_1^-$ lines from $V^* \setminus V_1^*$ fully below $\Delta_1$. The points $p_2(\xi_j)$ and $p_3(\xi_j)$ can be computed in the same way—this takes $O(n_2)$ and $O(n_3)$ time, respectively—after determining their $x$-coordinates.
in $O(1)$ time. (These $x$-coordinate are slope($\theta_{\xi_j} + \pi/3$) and slope($\theta_{\xi_j} + 2\pi/3$), respectively.) After computing $p_1(\xi_j)$, $p_2(\xi_j)$, and $p_3(\xi_j)$, we can make our decision: We proceed to the left if $p_1(\xi_j)$ lies below the line through $p_2(\xi_j)$ and $p_3(\xi_j)$, and we proceed to the right if $p_1(\xi_j)$ lies above that line. (In the fortunate situation that $p_1(\xi_j), p_2(\xi_j), p_3(\xi_j)$ are collinear, we can take $\bar{x} := \xi_j$, and we are done.)

Since each step in the binary search takes $O(n_1 + n_2 + n_3)$ time, the total binary search takes $O((n_1 + n_2 + n_3) \log n_1)$ time. Sorting the set $X$ before the binary search only increases this by a constant factor.

(iv) Finally, we take the two $x$-coordinates $\xi_i, \xi_{i+1}$ computed in the previous step, and find the points where $L_{med}$ crosses the vertical lines $x = \xi_i$ and $x = \xi_{i+1}$. Between $x = \xi_i$ and $x = \xi_{i+1}$, we know that $L_{med}$ lies inside a single trapezoid $r \in \Xi$. We then intersect $r$ with the slab $(\xi_i, \xi_{i+1}) \times (-\infty, \infty)$, to obtain the trapezoid $\Delta'_i$. (If, for example, we would have $(\xi_1, \xi_{i+1}) = (\xi_3, \xi_4)$ in Figure 7, then $\Delta'_i$ is the grey trapezoid.) Note that the number of lines crossing $\Delta'_i$ is at most $n_1/2$ and that the $x$-range $I'_i$ of $\Delta'_i$ satisfies property (P3).

After shrinking $\Delta_1$ in this manner, we proceed as follows. We first clip $\Delta_2$ so that its $x$-range corresponds to $I_1 \oplus \frac{\pi}{3}$, and then we shrink the clipped trapezoid $\Delta_2$ to a new trapezoid $\Delta'_2$ in the same way as we shrank $\Delta_1$ to $\Delta'_1$. Thus, $\Delta'_2$ is crossed by at most $n_2/2$ lines and $I'_2 \oplus \frac{\pi}{3}$ satisfies property (P3), where $I'_2$ is the $x$-range of $\Delta'_2$. Next, we clip the $x$-range of $\Delta_3$ to $I'_2 \oplus \frac{\pi}{3}$, and then we apply the shrinking procedure to $\Delta_3$ to obtain a new trapezoid $\Delta'_3$. Finally, we clip $\Delta'_1$ and $\Delta'_2$, so that their $x$-ranges correspond to $I'_3 \oplus \frac{\pi}{3}$ and $I'_3 \oplus \frac{\pi}{3}$, respectively. We then recurse on the triple $\Delta'_1, \Delta'_2, \Delta'_3$, passing along the appropriate sets $V'_i$ and updating the counts $n_i^{-}$.

The total time spent in all three shrinking steps is $O((n_1 + n_2 + n_3) \log(n_1 + n_2 + n_3))$, and each $n_i$ halves at every level in the recursion. Hence, the total time for Step 1 on page 9 is $O(n \log n)$. As already mentioned, Step 2 takes only constant time. We can conclude that we can find a collinear triple $p_1(\bar{x}), p_2(\bar{x}), p_3(\bar{x})$ in $O(n \log n)$ time. In the primal this corresponds to a triple of collinear concurrent lines as in Lemma 2.4, so we obtain the following theorem.

Theorem 2.6.

(i) We have $\beta^*_d = \sqrt{3}/2$. Moreover, for any multiset $V$ of $n$ voters in $\mathbb{R}^2$, we can compute a point $p$ with $\beta(p, V) \geq \sqrt{3}/2$ in $O(n \log n)$ time.

(ii) For $d \geq 3$, we have $1/\sqrt{d} \leq \beta^*_d \leq \sqrt{3}/2$. Moreover, for any multiset $V$ of $n$ voters in $\mathbb{R}^d$ with $d \geq 2$, we can compute a point $p$ with $\beta(p, V) \geq 1/\sqrt{d}$ in $O(n)$ time.

3 FINDING A POINT THAT MAXIMIZES $\beta(p, V)$

We know from Theorem 2.6 that, for any multiset $V$ of $n$ voters in $\mathbb{R}^d$, we can compute a point $p$ with $\beta(p, V) \geq 1/\sqrt{d}$ (even with $\beta(p, V) \geq \sqrt{3}/2$, in the plane). However, a given voter multiset $V$ may admit a $\beta$-plurality point for larger values of $\beta$—possibly even for $\beta = 1$. In this section, we study the problem of computing a point $p$ that maximizes $\beta(p, V)$, that is, a point $p$ with $\beta(p, V) = \beta(V)$, in the Real-RAM model.

3.1 An Exact Algorithm

Below, we sketch an exact algorithm to compute $\beta(V)$ together with a point $p$ such that $\beta(p, V) = \beta(V)$. Our goal is to show that, for constant $d$, this can be done in polynomial time. We do not make a special effort to optimize the exponent in the running time; it may be possible to speed up the algorithm, but it seems clear that it will remain impractical, because of the asymptotic running time, and also because of algebraic issues.
Note that we can efficiently check whether a true plurality point exists (i.e., \( \beta = 1 \) can be achieved) in time \( O(n \log n) \) by an algorithm of De Berg et al. [4], and if so, identify this point. Therefore, hereafter \( \beta = 1 \) is used as a sentinel value, and our algorithm proceeds on the assumption that \( \beta(p, V) < 1 \) for any point \( p \).

For a voter \( v \in V \), a candidate \( p \in \mathbb{R}^d \), and an alternative candidate \( q \in \mathbb{R}^d \), define \( f_v(p, q) := \min(|qv|/|pv|, 1) \) when \( p \neq v \), and define \( f_v(p, q) := 1 \) otherwise. Observe that for \( f_v(p, q) < 1 \), we have

- \( q \) wins voter \( v \) over \( p \) if and only if \( \beta > f_v(p, q) \).
- \( q \) and \( p \) have a tie over voter \( v \) if and only if \( \beta = f_v(p, q) \), and
- \( p \) wins voter \( v \) over \( q \) if and only if \( \beta < f_v(p, q) \).

For \( f_v(p, q) = 1 \), this is not quite true: When \( p = q = v \), we always have a tie, and when \( |pv| < |qv| \), then \( p \) wins \( v \) even when \( \beta = f_v(p, q) = 1 \). When \( p = q \) there is a tie for all voters, so the final conclusion (namely, that \( |V[p > \beta q]| \geq |V[p < \beta q]| \)) is still correct. The fact that we incorrectly conclude that there is a tie when \(|pv| < |qv|\) and \( \beta = f_v(p, q) = 1 \) does not present a problem either, since we assume \( \beta(p, V) < 1 \). Hence, we can pretend that checking if \( \beta > f_v(p, q) \), or \( \beta = f_v(p, q) \), or \( \beta < f_v(p, q) \) tells us whether \( q \) wins \( v \), or there is a tie, or \( p \) wins \( v \), respectively.

Hereafter, we identify \( f^+_v : \mathbb{R}^{2d} \to \mathbb{R} \) with its graph \( \{ (p, q, f_v(p, q)) \} \subset \mathbb{R}^{2d+1} \), which is a \( d \)-dimensional surface. Let \( f^+_v \) be the set of points lying above this graph, and \( f^-_v \) be the set of points lying below it. Thus, \( f^+_v \) is precisely the set of combinations of \( (p, q, \beta) \) where \( q \) wins \( v \) over \( p \), while \( f^-_v \) is the set where \( p \) ties with \( q \), and \( f^-_v \) is the set where \( q \) loses \( v \) to \( p \). Consider the arrangement \( \mathcal{A} := \mathcal{A}(F) \) defined by the set of surfaces \( F := \{ f^+_v : v \in V \} \). Each face of \( \mathcal{A} \) is a maximal connected set of points with the property that all points of \( C \) are contained in, lie below, or lie above, the same subset of surfaces of \( F \). (Note that we consider faces of all dimensions, not just full-dimensional cells.) Thus, for all \((p, q, \beta) \in C\), exactly one of the following holds: \( |V[p > \beta q]| < |V[p < \beta q]| \), or \( |V[p > \beta q]| = |V[p < \beta q]| \), or \( |V[p \beta q]| > |V[p < \beta q]| \). Let \( L \) be the union of all faces \( C \) of \( \mathcal{A}(F) \) such that \( |V[p > \beta q]| < |V[p < \beta q]| \), that is, such that \( p \) loses against \( q \) for all \((p, q, \beta) \in C\). We can construct \( \mathcal{A} \) and \( L \) in time \( O(n^{2d+1}) \) using standard machinery, as \( \mathcal{A} \) is an arrangement of degree-4 semi-algebraic surfaces of constant description complexity [2, 3]. We are interested in the set

\[
W := \{ (p, \beta) : |V[p > \beta q]| \geq |V[p < \beta q]| \text{ for any competitor } q \} \subset \mathbb{R}^{d+1}.
\]

What is the relationship between \( W \) and \( L \)? A point \((p, \beta)\) is in \( W \) precisely when, for every choice of \( q \in \mathbb{R}^d \), \( p \) wins at least as many voters as \( q \) (for the given \( \beta \)). In other words,

\[
W = \{ (p, \beta) : \text{there is no } q \text{ such that } (p, q, \beta) \in L \}.
\]

That is, \( W \) is the complement of the projection of \( L \) to the space \( \mathbb{R}^{d+1} \) representing the pairs \((p, \beta)\). The most straightforward way to implement the projection would involve constructing semi-algebraic formulas describing individual faces and invoking quantifier elimination on the resulting formulas [2]. Below, we outline a more obviously polynomial-time alternative.

Construct the vertical decomposition \( \text{vd}(\mathcal{A}) \) of \( \mathcal{A} \), which is a refinement of \( \mathcal{A} \) into pieces ("surfaces") \( \tau \), each bounded by at most \( 2(2d + 1) \) surfaces of constant degree and therefore of constant complexity; see Appendix A. A vertical decomposition is specified by ordering the coordinates— we put the coordinates corresponding to \( q \) last. Since \( \text{vd}(\mathcal{A}) \) is a refinement of \( \mathcal{A} \), the set \( L \) is the union of surfaces \( \tau \) of \( \text{vd}(\mathcal{A}) \) fully contained in \( L \). Since \( \mathcal{A} \) is an arrangement of \( n \) well-behaved surfaces in \( 2d + 1 \geq 5 \) dimensions, the complexity of \( \text{vd}(\mathcal{A}) \) is \( O(n^{2(2d+1)-4+\varepsilon}) = O(n^{4d-2+\varepsilon}) \), for any \( \varepsilon > 0 \) [21]. In particular, \( L \) comprises \( \ell := O(n^{4d-2+\varepsilon}) \) surfaces.
Since each \( \tau \subset L \) is a subface of the vertical decomposition \( \text{vd}(\mathcal{A}) \) in which the last \( d \) coordinates correspond to \( q \), the projection \( \tau' \) of \( \tau \) to \( \mathbb{R}^{d+1} \) is easy obtain (see Appendix A) in constant time; indeed it can be obtained by discarding the constraints on these last \( d \) coordinates from the description of \( \tau \). Thus, in time \( O(\ell) \), we can construct the family of all the projections of the \( \ell \) subfaces of \( L \), each a constant-complexity semi-algebraic object in \( \mathbb{R}^{d+1} \). We now construct the arrangement \( \mathcal{A}' \) of the resulting collection and its vertical decomposition \( \text{vd}(\mathcal{A}') \). The complexity of \( \text{vd}(\mathcal{A}') \) is either \( O(\ell^{d+1+\varepsilon}) \) or \( O(\ell^{2(d+1)-4+\varepsilon}) = O(\ell^{2d-2+\varepsilon}) \), depending on whether \( d + 1 \leq 4 \) or not, respectively [21]. Each subface in \( \text{vd}(\mathcal{A}') \) is either fully contained in the projection of \( L \) or fully disjoint from it. Collecting all of the latter subfaces, we obtain a representation of \( W \) as a union of at most \( O(\ell^{O(\varepsilon)}) = O(n^{O(d^2)}) \) constant-complexity semi-algebraic objects.

Now if \((p, \beta) \in W \) is the point with the highest value of \( \beta \), then \( \beta(V) = \beta(p, V) = \beta \). It can be found by enumerating all the subfaces of \( \text{vd}(\mathcal{A}') \) contained in the closure of \( W \)—we take the closure, because \( V(p, \beta) \) is defined as a supremum—and identifying their topmost point or points. Since each face has constant complexity, this can be done in \( O(1) \) time per subface.\(^5\) This completes our description of an \( O(n^{O(d^2)}) \)-time algorithm to compute the best \( \beta \) that can be achieved for a given set of voters \( V \), and the candidate \( p \) (or the set of candidates) that achieve this value. The above algorithm shows that the problem can be solved efficiently for constant \( d \). However, to the best of our knowledge, no polynomial time algorithm is known for the case when \( d \) is considered a variable.

### 3.2 An Approximation Algorithm

Since computing \( \beta(V) \) exactly appears expensive, we now turn our attention to approximation algorithms. In particular, given a voter set \( V \) in \( \mathbb{R}^d \) and an \( \varepsilon \in (0, 1/2) \), we wish to compute a point \( \tilde{p} \) such that \( \beta(\tilde{p}, V) \geq (1 - \varepsilon) \cdot \beta(V) \).

Our approximation algorithm works in two steps. In the first step, we compute a set \( P \) of \( O(n/\varepsilon^{2d-1} \log(1/\varepsilon)) \) candidates. \( P \) may not contain the true optimal point \( p \), but we will ensure that \( P \) contains a point \( \tilde{p} \) such that \( \beta(\tilde{p}, V) \geq (1 - \varepsilon/2) \cdot \beta(V) \). In the second step, we approximate \( \beta(p', V) \) for each \( p' \in P \), to find an approximately best candidate.

**Constructing the candidate set \( P \).** To construct the candidate set \( P \), we will generate, for each voter \( v_i \in V \), a set \( P_i \) of \( O(1/\varepsilon^{2d-1} \log(1/\varepsilon)) \) candidate points. Our final set \( P \) of candidates will be the union of the sets \( P_1, \ldots, P_n \). Next, we describe how to construct \( P_i \).

Partition \( \mathbb{R}^d \) into a set \( C \) of \( O(1/\varepsilon^{d-1}) \) simplicial cones with apex at \( v_i \) and opening angle \( \varepsilon/(2\sqrt{d}) \), so that for every pair of points \( u \) and \( u' \) in the same cone we have \( \angle u v_i u' \leq \varepsilon/(2\sqrt{d}) \). We assume for simplicity (and can easily guarantee) that no voter in \( V \) lies on the boundary of any of the cones, except for \( v_i \) itself and any voters coinciding with \( v_i \). Let \( C(v_i) \) denote the set of all cones in \( C \) whose interior contains at least one voter. For each cone \( C \in C(v_i) \), we generate a candidate set \( G_i(C) \) as explained next, and then we set \( P_i := \bigcup_{C \in C(v_i)} G_i(C) \cup \{v_i\} \).

Let \( d_C \) be the distance from \( v_i \) to the nearest other voter (not coinciding with \( v_i \)) in \( C \). Let \( A_1(C) \) be the closed spherical shell defined by the two spheres of radii \( \varepsilon \cdot d_C \) and \( d_C/\varepsilon \) around \( v_i \), as shown in Figure 8(i). The open ball of radius \( \varepsilon \cdot d_C \) is denoted by \( A_1^{\text{in}}(C) \), and the complement of the closed ball of radius \( d_C/\varepsilon \) is denoted by \( A_1^{\text{out}}(C) \). Let \( G_i(C) \) be the vertices in an exponential grid defined by a collection of spheres centered at \( v_i \), and the extreme rays of the cones in \( C \); see Figure 8(ii). The spheres have radii \((1 + \varepsilon/4)^i \cdot \varepsilon \cdot d_C \), for \( 0 \leq i \leq \log_{(1+\varepsilon/4)}(1/\varepsilon^2) = O((1/\varepsilon) \log(1/\varepsilon)) \). Observe

\(^5\)Once again, the projection to the \( \beta \) coordinate is particularly easy to obtain if, when constructing \( \text{vd}(\mathcal{A}') \), we set the coordinate corresponding to \( \beta \) first.
\(A_i(C)\) is a voter nearest to \(v_i\) defined by the two balls of radii \(\epsilon \cdot d_C\) and \(d_C / \epsilon\) around \(v_i\). The exponential grid \(G_i(C)\) is a collection of spheres centered at \(v_i\) plus extreme rays of the cones with apex at \(v_i\). The spheres have radii \((1 + \epsilon / 4)^i \cdot \epsilon \cdot d_C\) for \(0 \leq i \leq \log_4(1 + \epsilon / 4)(1 / \epsilon^2) = O((1 / \epsilon) \log(1 / \epsilon))\), and the interior angle of a cone is \(\epsilon / 2 \sqrt{d}\).

That \(G_i(C)\) contains not only points in \(C\), but in the entire spherical shell \(A_i(C)\). The set \(G_i(C)\) consists of \(O(1 / \epsilon^d \log(1 / \epsilon))\) points, and it has the following property:

Let \(p\) be any point in the spherical shell \(A_i(C)\), and let \(p'\) be a corner of the grid cell containing \(p\) and nearest to \(p\). Then, \(|p'p| \leq \epsilon \cdot |pv_i|\). (*)

To prove the property, let \(q\) be the point on the line containing \(pv_i\), on the same side of \(v_i\) as \(p\), such that \(|qp_i| = |p'v_i|\). From the construction of the exponential grid, we have \(|pq| \leq \frac{\epsilon}{4} \cdot |pv_i|\). Since \(p'\) and \(q\) lie in the same cone \(p'v_iq \leq \frac{\epsilon^3}{4}\) and, consequently, \(|p'q| \leq \frac{\epsilon}{4} \cdot |qp_i| \leq (1 + \frac{\epsilon}{4}) \cdot \frac{\epsilon}{2} \cdot |pv_i|\).

The property is now immediate, since \(|pp'| \leq |pq| + |qp'| < \epsilon \cdot |pv_i|\).

As mentioned above, \(P_i := \bigcup_{C \in C(v_i)} G_i(C) \cup \{v_i\}\), and the final candidate set \(P\) is defined as \(P := \bigcup_{v_i \in V} P_i\). Computing the sets \(P_i\) is easy: for each of the \(O(1 / \epsilon^{d-1})\) cones \(C \in C(v_i)\), determine the nearest neighbor of \(v_i\) in \(C\) in \(O(n)\) time by brute force, and then generate \(G_i(C)\) in \(O((1 / \epsilon^{d-1}) \log(1 / \epsilon))\) time. (It is not hard to speed up the nearest-neighbor computation using appropriate data structures, but this will not improve the final running time in Theorem 3.4.) We obtain the following lemma.

**Lemma 3.1.** The candidate set \(P\) has size \(O(n / \epsilon^{2d-1} \log(1 / \epsilon))\) and can be constructed in \(O(n^2 / \epsilon^{d-1} + n / \epsilon^{2d-1} \log(1 / \epsilon))\) time.

The next lemma is crucial to show that \(P\) is a good candidate set.

**Lemma 3.2.** For any point \(p \in \mathbb{R}^d\), there exists a point \(p' \in P\) with the following property: for any voter \(v_j \in V\), we have that \(|p'v_j| \leq (1 + 2\epsilon) \cdot |pv_j|\).

**Proof.** Let \(v_i\) be a voter nearest to \(p\). We will argue that the set \(P_i\) contains a point \(p'\) with the desired property. We distinguish three cases.

**Case I:** There is a cone \(C \in C(v_i)\) such that \(p\) lies in the spherical shell \(A_i(C)\). In this case, we pick \(p'\) to be a point of \(G_i(C)\) nearest to \(p\), that is, \(p'\) is a corner nearest to \(p\) of the grid cell containing \(p\). By property (*), we have

\[|p'v_j| \leq |p'p| + |pv_i| \leq \epsilon \cdot |pv_i| + |pv_j| \leq (1 + \epsilon) \cdot |pv_j|,\]

where the last inequality follows from the fact that \(v_i\) is a voter nearest to \(p\).
Case II: Point $p$ lies in $A_i^{ln}(C)$ for all $C \in C(v_i)$. In this case, we pick $p' := v_i$. Clearly $|p'v_j| = 0 \leq (1 + \epsilon) \cdot |pv_j|$ for $j = i$. For $j \neq i$, we argue as follows. Let $C \in C(v_i)$ be the cone containing $v_j$. Since we are in Case II, we know that $p \in A_i^{ln}(C)$, and so

$$|p'v_j| \leq |p'p| + |pv_j| \leq \epsilon d_C + |pv_j| \leq \epsilon |p'v_j| + |pv_j|. \quad (1)$$

Moreover, we have

$$|pv_j| \geq |p'v_j| - |pp'| \geq |p'v_j| - \epsilon d_C \geq |p'v_j|/2,$$  \quad (2)

where the last step uses that $\epsilon \leq 1/2$ and $d_C \leq |p'v_j|$. Combining (1) and (2), we obtain $|p'v_j| \leq (1 + 2\epsilon) \cdot |pv_j|$.

Case III: Cases I and II do not apply. In this case there is at least one cone $C$ such that $p \in A_i^{out}(C)$. Of all such cones, let $C^*$ be the one whose associated distance $d_C$ is maximized. Let $p''$ be the point on the segment $pv_j$ at distance $d_C/\epsilon$ from $v_i$. Without loss of generality, we will assume that $p$ and $v_i$ only differ in the $x_d$ coordinate; see Figure 9(i).

We will prove that the point $p''$ of $G_i(C^*)$ nearest to $p''$ (refer to Figure 9(i)) has the desired property. Consider a voter $v_j$. We distinguish three cases.

- When $i = j$, then we have

$$|p'v_j| \leq |p'p''| + |p''v_j| \leq (1 + \epsilon)|p''v_j| \leq (1 + \epsilon)|pv_j|,$$

where the second inequality follows from (*).

- When $v_j$ lies in a cone $C$ such that $p \in A_i^{ln}(C)$, then we can use the same argument as in Case II to show that $|p'v_j| \leq (1 + 2\epsilon) \cdot |pv_j|$.

- In the remaining case $v_j$ lies in a cone $C$ such that $p \in A_i^{out}(C)$. Let $v_k$ be a voter in $C$ nearest to $v_j$. Since $|v_i v_k| = d_C$, $|pv_i| \geq d_C/\epsilon$, and $|pv_k| \geq |pv_i|$, we can deduce that $\angle pv_i v_k \geq \pi/2 - \epsilon/2$, as illustrated in Figure 9(iii). Furthermore, since $v_k$ and $v_j$ belong to the same cone $C$ the angle $\angle v_k v_i v_j$ is bounded by $\epsilon/2 \sqrt{d} \leq \epsilon/2$ according to the construction. Putting the two angle bounds together, we conclude that $\angle pv_i v_j \geq \pi/2 - \epsilon$. Now consider the triangle defined by $p, v_i$ and $v_j$. From the Law of Sines, we obtain

$$\frac{|v_i v_j|}{\sin \angle v_i pv_j} = \frac{|pv_j|}{\sin \angle pv_i v_j}, \quad \text{or} \quad |v_i v_j| = |pv_j| \cdot \frac{\sin \angle pv_i v_j}{\sin \angle v_i pv_j} \leq \frac{|pv_j|}{\cos \epsilon} \leq (1 + \epsilon) \cdot |pv_j|.$$
for $\epsilon < 1/2$. Since $p''$ lies on the line between $p$ and $v_1$, we have
\[ |p''v_j| \leq \max(|pv_j|, |v_1v_j|) \leq (1 + \epsilon) \cdot |pv_j|. \]

Finally, we get the claimed bound by noting that $|p'p''| \leq \epsilon \cdot |p''v_j|$ (from (*)),
\[ |p'v_j| \leq |p'p''| + |p''v_j| \leq \epsilon \cdot |p''v_j| + (1 + \epsilon) \cdot |pv_j| \leq (1 + 2\epsilon) \cdot |pv_j|. \]

\[ \square \]

An approximate decision algorithm. Given a point $p$, two positive real values $\epsilon$ and $\beta$ and the voter multiset $V$, we say that an algorithm $\text{Alg}$ is an $\epsilon$-approximate decision algorithm if

- $\text{Alg}$ answers YES if $p$ is a $\beta$-plurality point, and
- $\text{Alg}$ answers NO if $p$ is not a $(1 - \epsilon)\beta$-plurality point.

In the remaining cases, where $(1 - \epsilon)\beta < \beta(p, V) < \beta$, $\text{Alg}$ may answer YES or NO.

Next, we propose an $\epsilon$-approximate decision algorithm $\text{Alg}$. The algorithm will use the so-called **Balanced Box-Decomposition (BBD)** tree introduced by Arya and Mount [1]. BBD trees are hierarchical space-decomposition trees such that each node $\mu$ represents a region in $\mathbb{R}^d$, denoted by region($\mu$), which is a $d$-dimensional axis-aligned box or the difference of two such boxes. A BBD tree for a set $P$ of $n$ points in $\mathbb{R}^d$ can be built in $O(n \log n)$ time using $O(n)$ space. It supports $(1 + \epsilon)$-approximate range counting queries with convex query ranges in $O(\log n + \epsilon^{-d})$ time [1]. In our algorithm all query ranges will be balls, hence a $(1 + \epsilon)$-approximate range-counting query for a $d$-dimensional ball $s(v, r)$ with center at $v$ and radius $r$ returns an integer $I$ such that $|P \cap s(v, r)| \leq I \leq |P| / |s(v, (1 + \epsilon)r)|$.

Our $\epsilon$-approximate decision algorithm $\text{Alg}$ works as follows.

1. Construct a set $Q$ of $O(n/\epsilon^{d-1})$ potential candidates competing against $p$, as follows. Let $Q(v)$ be a set of $O(1/\epsilon^{d-1})$ points distributed uniformly on the boundary of the ball $s(v, (1 - \epsilon/2) \cdot \beta \cdot |pv_j|)$, such that the distance between any point on the boundary and its nearest neighbor in $Q(v)$ is at most $\frac{1}{\sqrt{4d}} \cdot |pv_j| \leq \frac{1}{4} \cdot \beta \cdot |pv_j|$, as illustrated in Figure 10. In the last step, we use the fact that $\beta \geq 1/\sqrt{4d}$, according to Lemma 2.3. Set $Q := Q(v_1) \cup \cdots \cup Q(v_n)$.

2. Build a BBD tree $T$ on $Q$. Add a counter $c(\mu)$ to each node $\mu$ in $T$, initialized to zero.

3. For each voter $v \in V$ perform a $(1 + \epsilon/4)$-approximate range-counting query with $s(v, (1 - \epsilon/4) \cdot \beta \cdot |pv_j|)$ in $T$. We modify the search in $T$ slightly as follows. If an internal node $\mu \in T$ is visited and expanded during the search, then for every non-expanded child $\mu'$ of $\mu$ with region($\mu'$) entirely contained in $s(v, (1 - \epsilon/4) (1 - \epsilon/4) \cdot \beta \cdot |pv_j|)$ $\subseteq s(v, \beta \cdot |pv_j|)$, we increment the counter $c(\mu')$. Similarly, if a leaf is visited, then the counter is incremented if the point stored in the leaf lies within $s(v, (1 - \epsilon/4) \cdot \beta \cdot |pv_j|)$.

4. For a leaf $\mu$ in $T$, let $M(\mu)$ be the set of nodes in $T$ on the path from the root to $\mu$, and let $C(\mu) := \sum_{\mu' \in M(\mu)} c(\mu')$. Compute $C(\mu)$ for all leaves $\mu$ in $T$ by a pre-order traversal of $T$, and set $C := \max C(\mu)$.

5. If $C \leq n/2$, then return YES, otherwise NO.

To prove correctness of the algorithm, we define, for a given $\gamma > 0$, a fuzzy ball $s_\gamma(v, r)$ to be any set such that $s(v, r) \subseteq s_\gamma(v, r) \subseteq s(v, (1 + \gamma)r)$. Thus, if $q \in s(v, r)$, then $q \in s_\gamma(v, r)$, if $q \notin s(v, (1 + \gamma)r)$, then $q \notin s_\gamma(v, r)$, and otherwise $q$ may or may not be inside in $s_\gamma(v, r)$. We now observe that for each voter $v_1 \in V$ there is a fuzzy ball $s_{\epsilon/4}(v_1, (1 - \epsilon/4) \cdot \beta \cdot |pv_1|)$ such that the value $C(\mu)$ for a leaf $\mu$ storing a point $q$ is the depth of $q$ in the arrangement, denoted by $\mathcal{A}_{\epsilon/4}(V, 1 - \epsilon/4)$, of the fuzzy balls $s_{\epsilon/4}(v_1, (1 - \epsilon/4) \cdot \beta \cdot |pv_1|), \ldots, s_{\epsilon/4}(v_n, (1 - \epsilon/4) \cdot \beta \cdot |pv_n|)$.

**Lemma 3.3.** Algorithm $\text{Alg}$ $\epsilon$-approximately decides if $p$ is a $\beta$-plurality point in time $O(n/\epsilon^{d-1} \log \frac{\sqrt{d}}{\epsilon^{d-1}})$. 

Proof. We start by analyzing the running time of the algorithm. Constructing the set of points in \( Q \) can be done in time linear in \( |Q| \), while building the BBD-tree \( T \) requires \( O((n/e^{d-1}) \log(n/e^{d-1})) \) time [1, Lemma 1]. Next, the algorithm performs \( n \) approximate range queries, each requiring \( O(\log\frac{d}{\varepsilon^2} + \frac{1}{\varepsilon^2}) \) time [1, Theorem 2]). Note that the small modification we made to the query algorithm to update the counters does not increase the asymptotic running time. Finally, the traversal of \( T \) to compute \( C \) takes time linear in the size of \( T \), which is \( O(n/e^{d-1}) \).

It remains to prove that \( \text{Alg} \) is correct.

- If \( p \) is a plurality point, then there can be no point \( q \in \mathbb{R}^d \) having depth greater than \( n/2 \) in the arrangement of the balls \( s(v_1, \beta \cdot |pv|), \ldots, s(v_n, \beta \cdot |pv|) \). Since \( s_{\epsilon/4}(v, (1-\epsilon/4) \cdot \beta \cdot |pv|) \subseteq s(v, \beta \cdot |pv|) \), for all \( v \), \( \text{Alg} \) could not have found a point with depth greater than \( n/2 \), and hence, must return \text{YES}.

- If \( p \) is not a \((1-\epsilon)\beta\)-plurality point, then there exists a point \( q \) with depth greater than \( n/2 \) in the arrangement \( \mathcal{A}(V, 1-\epsilon) \) of the balls \( s(v_1, (1-\epsilon) \cdot \beta \cdot |pv|), \ldots, s(v_n, (1-\epsilon) \cdot \beta \cdot |pv|) \). Let \( q' \) be the point in \( Q \) nearest to \( q \). We claim that for any ball \( s(v, (1-\epsilon) \cdot \beta \cdot |pv|) \) that contains \( q \), its expanded version \( s(v, (1-\epsilon/4) \cdot \beta \cdot |pv|) \) contains \( q' \). Of course, if \( s(v, (1-\epsilon) \cdot \beta \cdot |pv|) \) contains \( q' \), then we are done. Otherwise, let \( x \) be the point where \( qq' \) intersects the boundary of \( s(v, (1-\epsilon) \cdot \beta \cdot |pv|) \); see Figure 10. Note that \( q' \) must also be the point in \( Q \) nearest to \( x \). Let \( x' \) be the point on the boundary of \( s(v, (1-\epsilon/4) \cdot \beta \cdot |pv|) \) nearest to \( x \), and let \( q'' \) be a point in \( Q \) on the boundary of \( s(v, (1-\epsilon/2) \cdot \beta \cdot |pv|) \). By construction, we have

\[
|xx'| = \frac{\varepsilon}{4} \cdot \beta \cdot |pv| \quad \text{and} \quad |x'q''| \leq \frac{\varepsilon}{4} \cdot \beta \cdot |pv|
\]

and, by the triangle inequality, we obtain

\[
|xq'| \leq |xq''| \leq |xx'| + |x'q''| \leq \frac{\varepsilon}{2} \cdot \beta \cdot |pv|.
\]

This implies that \( s(v, (1-\epsilon/4) \cdot \beta \cdot |pv|) \subseteq s_{\epsilon/4}(v, (1-\epsilon/4) \cdot \beta \cdot |pv|) \) must contain \( q' \). Consequently, if \( q \) has depth at least \( n/2 \) in \( \mathcal{A}(V, 1-\epsilon) \) then \( q' \) has depth at least \( n/2 \) in the arrangement \( \mathcal{A}_{\epsilon/4}(V, (1-\epsilon/4)) \), and hence, the algorithm will return \text{NO}. \( \square \)
The algorithm. Now, we have the tools required to approximate $\beta(V)$. First, generate the set $P$ of $O\big(\frac{n^3}{\varepsilon^d} \log \frac{1}{\varepsilon} \big)$ candidate points (Lemma 3.1). For each candidate point $p \in P$, perform a binary search for an approximate $\beta^*(p)$ in the interval $[1/\sqrt{d}, 1]$, until the remaining search interval has length at most $\varepsilon/2 \cdot 1/\sqrt{d}$. For each $p$ and $\beta^*$, $(\varepsilon/2)$-approximately decide if $p$ is a $\beta^*$-plurality point in $V$ using ALG (Lemma 3.3). Return the largest $\beta^*$ and the corresponding point $p$ on which the algorithm says YES.

**Theorem 3.4.** Given a multiset $V$ of voters in $\mathbb{R}^d$, a $((1-\varepsilon) \cdot \beta(V))$-plurality point can be computed in $O\big(\frac{n^3}{\varepsilon d^2} \cdot \log \frac{n}{\varepsilon^2 \varepsilon} \cdot \log^2 \frac{1}{\varepsilon}\big)$.

## 4 CONCLUDING REMARKS

We proved that any finite set of voters in $\mathbb{R}^d$ admits a $\beta$-plurality point for $\beta = 1/\sqrt{d}$ and that some sets require $\beta = \sqrt{3}/2$. For $d = 2$, we managed to close the gap by showing that $\beta^*_{2} = \sqrt{3}/2$. One of the main open problems is to close the gap for $d > 2$. Recall that recently the bounds for $d \geq 4$ have been improved—see footnote 3 in the Introduction—but there is still a small gap left between the upper and lower bound. We also presented an approximation algorithm that finds, for a given $V$, a $(1-\varepsilon) \cdot \beta(V)$-plurality point. The algorithm runs in $O^*(n^2/\varepsilon^{3d-2})$ time. Another open problem is whether a subquadratic approximation algorithm exists, and to prove lower bounds on the time to compute $\beta(V)$ or $\beta(p,V)$ exactly. Finally, it will be interesting to study $\beta$-plurality points in other metrics, for instance in the personalized $L_1$-metric [4] for $d > 2$ or in the $L_1$-metric for $d \geq 2$.

## APPENDIX

### A A PRIMER ON VERTICAL DECOMPOSITIONS

We follow the notation and terminology of References [7, 26]. A vertical decomposition is, roughly, any partition of space into finitely many so-called cylindrical cells (see below for a definition); it need not be a topological complex. A vertical decomposition of an arrangement is a refinement of an arrangement into cylindrical cells, where refinement means that each cylindrical cell is a subset of a face in the arrangement. We define cylindrical cells recursively. To simplify the notation, any inequality limit in our definitions can be omitted, i.e., replaced by a $\pm \infty$, as appropriate. For example, when we talk about an open interval $(a, b)$, i.e., the set of numbers $x$ with $a < x < b$, we include the possibilities of the unbounded intervals $(\pm \infty, b)$, $(a, +\infty)$, and $(-\infty, +\infty)$.

A one-dimensional cylindrical cell is either a singleton or an open interval $(a, b)$. So a one-dimensional vertical decomposition is a decomposition of $\mathbb{R}$ into a finite number of singletons and intervals.

We now define a cylindrical cell $\tau$ in $\mathbb{R}^2$. Its projection $\tau'$ to the $x_1$-axis is a cylindrical cell in $\mathbb{R}$. The cell $\tau$ must have one of the following two forms:

- $\{ (x_1, f_2(x_1)) \mid x_1 \in \tau' \}$, where $f_2 : \tau' \to \mathbb{R}$ is a continuous total function, or
- $\{ (x_1, x_2) \mid x_1 \in \tau', f_2(x_1) < x_2 < g_2(x_1) \}$, where $f_2, g_2 : \tau' \to \mathbb{R}$ are two continuous total functions, with the property that $f_2(x_1) < g_2(x_1)$ for all $x_1 \in \tau'$.

If $\tau'$ is a singleton, then the former defines a vertex and the latter an (open) vertical segment. If $\tau'$ is an interval, then the former defines an open monotone arc (a portion of the graph of the function $f_2$) and the latter an open pseudo-trapezoid delimited by two (possibly degenerate) vertical

---

6The specific decomposition depends on the algorithm used to construct it and on the ordering of the coordinates. In the computational- and combinatorial-geometry literature, one often speaks of "the vertical decomposition of the arrangement" in the sense of "the vertical decomposition obtained by applying the algorithm, say, of Chazelle et al. [7] or of Koltun [21], to the given arrangement."
segments on left and right and by the two disjoint function graphs below and above. (Recall that any of the limits may be omitted. For example, \( \mathbb{R}^2 \) is a legal cell in a trivial two-dimensional vertical decomposition consisting only of itself, where all the limits have been “replaced by infinities.”)

A cylindrical cell \( \tau \subset \mathbb{R}^d \) is defined recursively. Its projection \( \tau' \) is a cylindrical cell in \( \mathbb{R}^{d-1} \). Moreover, \( \tau \) must have one of the following forms:

- \( \{(x', f_d(x_1, \ldots, x_{d-1})) \mid x' \in \tau'\}, \) where \( f_d : \tau' \to \mathbb{R} \) is a continuous total function, or
- \( \{(x', x_d) \mid x' \in \tau', f_d(x') < x_d < g_d(x')\}, \) where \( f_d, g_d : \tau' \to \mathbb{R} \) are two continuous total functions, with the property that \( f_d(x') < g_d(x') \) for all \( x' \in \tau' \).

A cylindrical cell is fully specified by giving its dimension and the sequence of functions \( f_i \) or pairs of functions \( f_i, g_i \), as appropriate. In particular, the projection of the cell in a \( k \)-dimensional decomposition to its first \( k' < k \) coordinates can be obtained by retaining the inequalities in the first \( k' \) coordinates and discarding the remaining ones.

**ACKNOWLEDGMENT**

The authors thank Sampson Wong for improving an earlier version of Lemma 2.4.

**REFERENCES**


