MASTER

Improved Bounds for Discrete Voronoi Games

van Wordragen, G.C.

Award date:
2022

Department of Mathematics and Computer Science
Algorithms, Geometry & Applications

Improved Bounds for Discrete Voronoi Games

Master's Thesis

Geert van Wordragen

Supervision:
prof. dr. Mark de Berg
prof. dr. Frits Spieksma

Assessment committee:
prof. dr. Mark de Berg
prof. dr. Frits Spieksma
dr. Jaron Sanders
dr. ir. Tim Willemse

July 8, 2022

Abstract

In the one-round discrete Voronoi game, two players compete over a set of voters represented by points $V \subset \mathbb{R}^d$. They do this by placing points: Player 1 places k points, followed by Player 2 placing ℓ, after which each voter is won by the player that placed a point closest to it. For three different definitions of 'closest', we present lower bounds on the number of voters Player 1 wins under optimal play of the game in $\mathbb{R}^2$ with $\ell = 1$. Such bounds already existed when 'closest' is based on the L_2 metric, so there we prove better bounds with an algorithm based on the quadtree of V. We also introduce bounds for the L_1 metric and for the personalised L_1 metric, where each voter has a preference vector describing how they value the distance in each coordinate.

Contents

1 Introduction **3**
1.1 Related work 4
1.2 Contributions and organisation 5

2 Solutions for low k **7**
2.1 ε-nets 7
2.2 L_2 metric 8
2.3 L_1 metric 10
2.4 Personalised L_1 metric 11
2.4.1 Preference-oblivious methods 12
2.4.2 Preference-aware methods 14

3 Solutions for high k **17**
3.1 Algorithm 17
3.2 A first bound 19
3.3 A tight bound 21
3.4 A more refined approach 23
3.4.1 Incorporating techniques for low k 24
3.4.2 Excluding quadrants 25
3.5 L_1 metric 26
3.5.1 A more refined approach 27

4 Conclusion **29**
4.1 Future work 29
4.1.1 Higher dimensions 30
4.1.2 VG(k, ℓ) with $\ell > 1$ 30
4.1.3 Algorithms to achieve the found bounds 30
4.1.4 β-plurality 31
4.1.5 Measures 31

Chapter 1

Introduction

Two-player discrete Voronoi games are a way of modelling elections in a two-party system. As input, we get a set of voters with points in $\mathbb{R}^d$ to represent their opinions on d topics. The two competing players then decide what standpoints their parties should take, again represented by points in $\mathbb{R}^d$. After that the voters will vote on the party whose standpoints are the closest to their own opinion. The goal for each player is to win as many votes as possible. For Player 1, the problem is closely related to that of finding a plurality point; a point Player 1 can place to guarantee that with one point Player 2 will win at most half of the voters.

Another way to interpret Voronoi games is as a competitive facility location problem, where the goal is to find locations to build facilities that are in some sense good. In a (non-discrete) Voronoi game, the competitors place their facilities as points in $\mathbb{R}^2$ and a facility placement is good if for a large region it is the closest facility. We can determine this by drawing the Voronoi diagram for all points and looking at the area covered by each of the Voronoi cells. In discrete Voronoi games, we specifically consider where the users of these facilities live: as input we get a set of points in $\mathbb{R}^2$ representing the users, and the score for a player depends on how many users live in the player's Voronoi cells. Figure 1.1 gives an example of a situation after Player 1 (red) and Player 2 (green) have placed their points. The blue crosses are users, so here Player 1's one facilities are the closest for eight users and Player 2's for three.

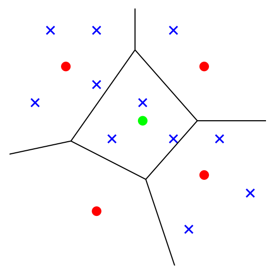


Figure 1.1: Discrete Voronoi game after both players have placed their points

We call a (discrete) Voronoi game a *one-round game* if all of Player 1's points are placed before Player 2's. This setting was introduced by Banik *et al.* [4], who use $\mathrm{VG}(k, \ell)$ to denote the one-round discrete Voronoi game where Player 1 places k points and Player 2 places ℓ. The setup means Player 1 only knows where the voters are and needs to account for every possible placement of Player 2's points, while Player 2 has complete information. As a consequence, the game is harder for Player 1, both in computational complexity and in chance of winning.

This thesis will consider, given k and ℓ, how many voters Player 1 can always win regardless of how the voters are distributed. Note that this is equivalent to asking how many points Player 1 needs to place in order to win a given number of voters.

1.1 Related work

Algorithms to find an optimal placement. The first question that might come to mind is what Player 1 should do to win as many voters as possible from a given set V. The problem $\mathrm{VG}(1,1)$ in $\mathbb{R}$ can be solved by finding the median, for which Blum *et al.* [10] have found a linear-time algorithm. When we instead work in $\mathbb{R}^d$ (with the L_2 metric), the problem is equal to that of finding the Tukey median, a generalisation of the median to higher dimensions that can be found in $\mathcal{O}\left(n^{d-1}+n\log n\right)$ time [13]. We can also look at the problem $\mathrm{VG}(k,\ell)$ and remain in $\mathbb{R}$. In this situation, De Berg *et al.* [9] give an algorithm that solves it in $\mathcal{O}\left(kn^4\right)$ time. On the other hand, they also show that in $\mathbb{R}^2$ the decision problem is already $\sum_2^P$-hard, which means it is NP-hard and likely not in NP. They do give an algorithm that solves the most general problem ($\mathrm{VG}(k,\ell)$ in $\mathbb{R}^d$ with L_p metric) optimally in $\mathcal{O}\left((n^ck\ell)^{(kd+1)(\ell d+1)}p^{k\ell d^2}\right)$ time, for some constant c.

	$\mathbb{R}$	$\mathbb{R}^d$
$\mathrm{VG}(1,1)$	n [10]	$n^{d-1}+n\log n$ (for L_2 only) [13]
$\mathrm{VG}(k,\ell)$	kn^4 [9]	$(n^ck\ell)^{(kd+1)(\ell d+1)}p^{k\ell d^2}$ (for L_p) [9]

Table 1.1: Lowest known big-O running time to get an optimal solution

Van Hulzen [19] gives heuristic algorithms to still quickly find a solution in $\mathbb{R}^2$, and also a way to find the optimal strategy for Player 2 in $\mathcal{O}\left(n^{2\ell}+n^5\right)$ time. Banik *et al.* [5] give algorithms to find the optimal placement for both players in the situation that points have already been placed and each needs to place one more, in $\mathbb{R}^2$ with the L_2 but also the L_1 and the L_∞ metric.

Bounds on the number of voters Player 1 can win. Another interesting question is how many voters Player 1 will win, assuming both players play optimally. Only when Player 2 places a single point are the bounds for this known, and these are only known to be tight for $\mathrm{VG}(1,1)$. There we can use the bounds Chawla *et al.* [14] have found for the Tukey median. They showed that for any voter set in $\mathbb{R}^d$ the optimal score for Player 1 will be in $[\frac{1}{d+1},\frac{1}{2}]$, and conversely any value in that interval is the optimal score for some voter set.

For $\mathrm{VG}(k,1)$ bounds are given by Banik *et al.* [6]. They give an upper bound of $1-\frac{1}{2k}$ for the number of voters Player 1 wins in optimal play, while giving two separate lower bounds. One lower bound, which works best when k is relatively low, is found by constructing an ε-net over the voter set. There is no closed form expression for the resulting lower bounds, but Table 1.2 shows some values in $\mathbb{R}^2$. Finding such an ε-net can be done in $\mathcal{O}\left(kn\log^4(n)\right)$ time. What ε-nets are and why they work will be explained further in Chapter 2. In $\mathbb{R}^2$, Banik *et al.* also prove a lower bound of $1 - \frac{42}{k}$, which is the better bound when $k \geq 137$. The accompanying algorithm takes $\mathcal{O}\left(n^2\right)$ time, or more if a more space-efficient method is used for finding the k-enclosing disk. In $\mathbb{R}^3$, the alternative lower bound of $1 - \frac{420}{k}$ is only better for $k \geq 806$.

Plurality points. Lastly, one might only be interested in Player 1 winning the game, meaning all situations where Player 1 wins less voters than Player 2 are equally bad. This has been studied for $\mathrm{VG}(1,1)$, where a point Player 1 can place to win is referred to as a plurality point. De Berg *et al.* [8] give an $\mathcal{O}(dn \log n)$ algorithm for $\mathbb{R}^d$ to find such a plurality point or return that there is none (in the L_2-norm). They also give an $\mathcal{O}(n)$ algorithm for the L_1-norm and an $\mathcal{O}\left(n^{d-1}\right)$ algorithm for the *personalised* L_1*-norm*, where each voter has a preference vector that describes how important they deem the distance in each coordinate.

Often, there is no plurality point. One solution by Aronov *et al.* [3] to this is to instead look at β-plurality, where a voter's distance to Player 1's point is multiplied by β while the distance to Player 2's point remains the same. This is a way of modelling *valence*; factors such as charisma, competence and campaign spending. The lower β, the higher the valence of Player 1's party compared to Player 2's. The paper gives algorithms to find a β-plurality point with β as high as possible and gives bounds on how low this β can need to be.

Continuous Voronoi games. Continuous Voronoi games take place on a region in $\mathbb{R}^d$, where each point's score is determined by how much of the region lies in its Voronoi cell, instead of how many voters. For continuous Voronoi games where both players place the same number of points, the one-dimensional game can always be won by Player 1 if it is one-round and by Player 2 otherwise [1]. In $\mathbb{R}^2$, Player 2 will win the one-round game as long as n is large enough, in both the L_2 metric [15] and the L_1 metric [12].

1.2 Contributions and organisation

This thesis considers the one-round discrete Voronoi game $\mathrm{VG}(k,1)$ in $\mathbb{R}^2$ with the L_2, L_1 and personalised L_1 metrics. It gives new lower bounds on the number of voters Player 1 can win and algorithms to ensure those lower bounds. More formally, we derive bounds on the quantity $\Gamma(n,k)$ defined as follows:

$$\Gamma(n,k) = \min\{\Gamma(V,k) \mid V \text{ is a set of } n \text{ voters in } \mathbb{R}^2\},$$

where $\Gamma(V,k)$ denotes the maximum number of voters that Player 1 can win from the voter set V when placing a set P of k points against an opponent that optimally places a single point q. Player 2 wins a voter when q is strictly closer to it than all points $p \in P$; otherwise Player 1 wins it.

No papers have been written yet on bounds for the L_1 and personalised L_1 metrics in one-round discrete Voronoi games, so there they are new. For the L_2 metric the bounds by Banik *et al.* [6] are improved upon. For example, we are able to win half of the voters using only four (instead of five) points, and also a fraction $1 - \frac{20}{k}$ of the voters voters (instead of $1 - \frac{42}{k}$) with k points.

Chapter 2 considers strategies based on ε-nets that work well when Player 1 places a (relatively) low number of points. For the L_2 metric this means we find better ε-nets for convex sets, for the L_1 metric we prove that we can use ε-nets for axis-parallel rectangles, and for the personalised L_1 metric we prove we can use ε-nets for axis-parallel star-shaped sets and show how to construct these. The results for this up to $k = 5$ and the previous results from Banik *et al.* [6] are given in Table 1.2.

k	1	2	3	4	5
L_2 (Banik *et al.* [6])	$n/3$	$3n/7$	$7n/15$	$15n/31$	$21n/41$
L_2 (new)				$n/2$	$11n/21$
L_1	$n/2$	$3n/5$	$2n/3$	$5n/7$	$3n/4$
Personalised L_1		$n/4$	$2n/7$	$n/3$	$4n/11$

Table 1.2: Lower bounds for $\Gamma(n, k)$

Chapter 3 gives a quadtree-based strategy that works well when Player 1 places a relatively large number of points. For the L_2 metric this gives a bound of $1 - \frac{20}{k}$ and for the L_1 metric it gives a bound of $1 - \frac{6\frac{6}{7}}{k}$. In both cases, there is also an $\mathcal{O}(n \log n)$ algorithm that accomplishes a slightly worse bound.

Chapter 2

Solutions for low k

As shown by Banik *et al.* [6], ε-nets can provide a solution for Player 1. These solutions are mostly interesting for low k, due to there being better techniques for higher k that do not only consider solutions in the form of ε-nets.

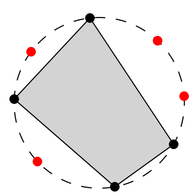
2.1 ε-nets

To properly define what such an ε-net exactly is, we first need to define range spaces.

Definition 2.1.1. *A* range space *is a pair* $(X, \mathcal{R})$ *where* $\mathcal{R}$ *is a family of subsets of* X*. Elements of* X *are referred to as* points *and elements of* $\mathcal{R}$ *as* ranges*.*

For our purposes, X will always simply be $\mathbb{R}^2$ and $\mathcal{R}$ can be the family of axis-parallel rectangles or the family of convex sets. Other commonly used ranges are for example disks and half-spaces.

Definition 2.1.2. *An* ε-net *for a finite set of points* $P \subset X$ *from a range space* $(X, \mathcal{R})$ *is a set of points* $N \subset X$ *such that any range* $R \in \mathcal{R}$ *where* $|R \cap P| \geq \varepsilon |P|$ *must also contain a point of* N*. The* ε*-net is called* strong *when* $N \subseteq P$*.*



A good ε-net will contain only a small number of points, making it useful for example for approximation. To get a strong ε-net, we can simply pick random points from P. For many range spaces this already gives a good result with high probability [18]. However, when as ranges we have the convex sets, strong ε-nets that are small do not exist. For example, when the points of P lie on a circle, we can always make a convex set that covers all points except those chosen for the ε-net, by taking their convex hull. Therefore, we need to use 'weak' ε-nets to get non-trivial bounds there.

A simple example of a weak $\frac{2}{3}$-net is the *centerpoint*. This is a point such that any line going through it divides the point set roughly equally: any side of that line will contain at least $\frac{1}{3}$ of the points. It can always be found in linear time [20]. Any convex set that does not contain the centerpoint must lie completely on one side of some line through the centerpoint, and therefore must contain less than $\frac{2}{3}$ of the points.

In general, we will use ε_k to refer to the lowest value such that k points suffice to make an ε_k-net for any point set, and $\overline{\varepsilon_k}$ the lowest value that has been proven we can reach. We will use $\varepsilon_k^{\mathcal{C}}$ when the ranges are convex sets and $\varepsilon_k^{\square}$ when the ranges are axis-parallel rectangles.

2.2 L_2 metric

When working with the L_2 metric, we can use ε-nets for convex sets to get upper bounds on $V(k, 1)$. This is because the Voronoi cell corresponding to the point Player 2 places is convex and does not contain any points of Player 2. Banik *et al.* [6] use the results by Mustafa and Ray [21], but the following section will show a construction that gives better bounds.

Let V be a point set in general position in the plane and $L = \{\ell_1, \ell_2, \ell_3\}$ be a set of three concurrent lines. The lines in L partition the plane into six wedges. We define the *weight* of a wedge to be the number of points from V in the wedge's interior, plus $\frac{1}{2}$ for each point on the wedge's boundary. The following theorem is a generalisation of the proposition from Bukh [11] that three concurrent lines can equipartition a point set.

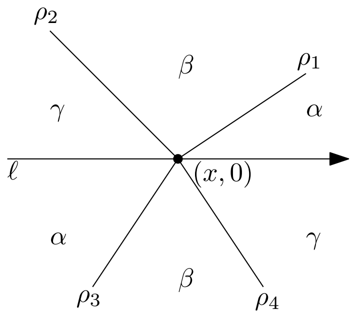


Figure 2.1: Situation for Theorem 2.2.1

Theorem 2.2.1. *Let V be a point set in general position and $\alpha, \beta, \gamma \in \mathbb{N}_0$ such that $2\alpha + 2\beta + 2\gamma = |V|$. Then there is a set of three concurrent lines that partition the plane into six wedges with weights (in counterclockwise order) $\alpha, \beta, \gamma, \alpha, \beta, \gamma$.*

Proof. Let θ be an arbitrary angle and ℓ the directed line that makes that angle θ with the positive x-axis and also splits V in half, i.e. it has exactly $|V|/2$ points on either side of it. Because V is in general position such a line exists for any θ. Without loss of generality, assume ℓ is the x-axis and consider a point $(x, 0) \in \ell$.

Let $\rho(\phi)$ be the ray emanating from $(x, 0)$ whose counterclockwise angle with ℓ is ϕ and $F(\phi)$ denote the weight of the wedge between ℓ and $\rho(\phi)$. We have $F(0) = 0$ and $F(\pi) = \frac{|V|}{2}$. Since V is in general position, ρ will always go through at most two points, thus as ϕ goes from 0 to π, $F(\rho)$ increases in steps of either $\frac{1}{2}$ followed by $\frac{1}{2}$, or one followed by one. Therefore, there are always rays $\rho_1 = \rho(\phi_1)$ such that $F(\phi_1) = \alpha$. Likewise, there are rays ρ_2, ρ_3, ρ_4 that together give six wedges with the given weights. Figure 2.1 depicts this situation.

Now imagine that we move the point $(x, 0)$ by changing the value of x. Consider what happens to the rays $\rho_1, \ldots, \rho_4$ that give the requested wedges. As we decrease x all rays will move towards ℓ and as we increase x they will move towards $-\ell$. This movement is continuous, so there is a value x_1 where ρ_1 and ρ_3 line up, as well as a value x_2 where ρ_2 and ρ_4 line up.

These values again change continuously as we change θ. When we add π to θ, we get the same picture but with the rays swapped around: ρ_1 and ρ_3 switch roles, as do ρ_2 and ρ_4. The points $(x_1, 0)$ and $(x_2, 0)$ will appear in the same position, but this means that with respect to ℓ they have changed order. Thus, for some θ they must have been equal.

$\square$

Combined with Lemma 2.2.2 this gives us the tools to create new ε-nets.

Lemma 2.2.2. *Let L be a set of three lines intersecting in a common point p, and consider the six wedges defined by L. A convex set S not containing p will always intersect at most four wedges. These wedges must be adjacent.*

Proof. If S intersects two wedges, then it needs to also intersect the ray(s) between those wedges. Because S is convex it cannot intersect both rays that make up a line without also intersecting their shared point p. Thus, S can intersect at most three of the rays, giving at most four adjacent wedges. $\square$

If we simply let $\alpha = \beta = \gamma = \frac{1}{6}|V|$, the lemma lets us conclude that S overlaps at most $\frac{2}{3}$ of the points in V. This is in line with the centerpoint theorem.

We can now use the partition into six wedges to get better bounds for $k \geq 4$ on $\varepsilon_k^{\mathcal{C}}$, the effectiveness of a k-point ε-net for convex ranges.

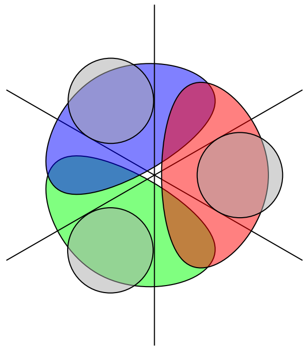


Figure 2.2: Sets of wedges

Theorem 2.2.3. *Let $\varepsilon_k^{\mathcal{C}}$ be the lowest value for the range space of convex sets in $\mathbb{R}^2$ such that we can make a k-point $\varepsilon_k^{\mathcal{C}}$-net for any set of points. Given $r_1, r_2, r_3, s \in \mathbb{N}_0$,*

$$\varepsilon_{r_1+r_2+r_3+3s+1}^{\mathcal{C}} \leq \frac{1}{2}\left(\frac{1}{\varepsilon_{r_1}^{\mathcal{C}}} + \frac{1}{\varepsilon_{r_2}^{\mathcal{C}}} + \frac{1}{\varepsilon_{r_3}^{\mathcal{C}}}\right)^{-1} + \frac{1}{2}\varepsilon_s^{\mathcal{C}}.$$

Proof. Let $\mu = \frac{1}{2}\left(\frac{1}{\varepsilon_{r_1}^{\mathcal{C}}} + \frac{1}{\varepsilon_{r_2}^{\mathcal{C}}} + \frac{1}{\varepsilon_{r_3}^{\mathcal{C}}}\right)^{-1}$ and assume we are given n points. If n is odd, we add an arbitrary extra point to the set (but not to n), so that we can use Theorem 2.2.1 with $\alpha = \left\lceil \frac{\mu}{\varepsilon_{r_1}^{\mathcal{C}}} n \right\rceil$, $\beta = \left\lceil \frac{\mu}{\varepsilon_{r_2}^{\mathcal{C}}} n \right\rceil$ and $\gamma = \left\lceil \frac{n}{2} - \alpha - \beta \right\rceil$ to get a partition, and place the point p at the intersection of the lines. We then place an $\varepsilon_s^{\mathcal{C}}$-net for each of the three sets of three wedges shown in Figure 2.2, an $\varepsilon_{r_1}^{\mathcal{C}}$-net for the wedge with size α, an $\varepsilon_{r_2}^{\mathcal{C}}$-net for the wedge with size β and an $\varepsilon_{r_3}^{\mathcal{C}}$-net for the wedge with size γ.

From Lemma 2.2.2 we know that a convex set not containing p can intersect at most four adjacent wedges. Thus, it will always intersect three wedges with an $\varepsilon_s^{\mathcal{C}}$-net, from which it thus gets less than $\frac{n}{2}\varepsilon_s^{\mathcal{C}}$ points, and one wedge with its own ε-net, from which it gets less than $\varepsilon_{r_1}^{\mathcal{C}} \left\lceil \frac{\mu}{\varepsilon_{r_1}^{\mathcal{C}}} n \right\rceil$, $\varepsilon_{r_2}^{\mathcal{C}} \left\lceil \frac{\mu}{\varepsilon_{r_2}^{\mathcal{C}}} n \right\rceil$ or $\varepsilon_{r_3}^{\mathcal{C}} \left\lceil \frac{n}{2} - \alpha - \beta \right\rceil$ points, thus at most μn. $\square$

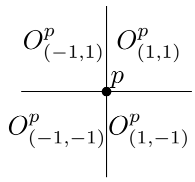
Perhaps the most interesting consequence of this is the Corollary 2.2.4. For us, it means that Player 1 always only needs to place four points to win at least as many voters as Player 2, as opposed to the five that were proven in earlier work. This is also something Van Hulzen [19] noticed in practice.

Corollary 2.2.4. $\varepsilon_4^{\mathcal{C}} \leq \frac{1}{2}$.

Proof. This follows from Theorem 2.2.3 with $r_1, r_2, r_3 = 0$ and $s = 1$. □

2.3 L_1 metric

Voronoi cells under the L_1 metric are not convex but only star-shaped, so we cannot use ε-nets in the same way as before. Instead, we can use the following very useful lemma.



Lemma 2.3.1. *Let $p \in \mathbb{R}^d$ be a point played by Player 1. For a vector $c \in \{-1, 1\}^d$, define the orthants*

$$O_c^p = \left\{ x \in \mathbb{R}^d \mid c_i(x_i - p_i) \geq 0 \text{ for all } 1 \leq i \leq d \right\}.$$

A point q by Player 2 that wins a voter $v \in V \cap O_c^p$ must have $\langle c, q \rangle > \langle c, p \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product.

Proof. Let $v \in V \cap O_c^p$. For any i we know $c_i(v_i - p_i) \geq 0$ and $|c_i| = 1$, therefore $c_i(v_i - p_i) = |v_i - p_i|$ and

$$\|v - p\|_1 = \sum_{i=1}^{d} |v_i - p_i| = \sum_{i=1}^{d} c_i(v_i - p_i) = \langle c, v - p \rangle.$$

Because $|c_i| = 1$ for all i we also know that

$$\|v - q\|_1 = \sum_{i=1}^{d} |v_i - q_i| \geq \sum_{i=1}^{d} c_i(v_i - q_i) = \langle c, v - q \rangle.$$

If we assume q wins v, then $\|v - q\|_1 < \|v - p\|_1$. This now gives us

$$\langle c, v - q \rangle \leq \|v - q\|_1 < \|v - p\|_1 = \langle c, v - p \rangle,$$

which implies $\langle c, q \rangle > \langle c, p \rangle$. □

A direct consequence of this lemma is that there is no vector c where Player 2 wins voters from two opposite orthants O_c^p and O_{-c}^p. In $\mathbb{R}^2$ this means Player 2 can only win voters from the two quadrants on the left, right, top, or bottom of a point p played by Player 1. In higher dimensions there are more combinations of orthants possible so this does not work out as nicely.

Theorem 2.3.2. *Let V be a set of n voters in $\mathbb{R}^2$. If Player 1 places points according to an ε-net for V with respect to axis-parallel rectangles, then Player 2 always wins less than εn voters.*

Proof. As noted, Player 2 has to choose for each point of Player 1 whether they want to win voters on the left, right, top, or bottom of that point. Combining these constraints, we get that the voters won by Player 2 must lie in an (open) axis-parallel rectangle that does not contain any point of Player 1. Because Player 1's points are placed according to an ε-net, this means any axis-parallel rectangle not containing these cannot contain εn voters. □

Aronov *et al.* [2] give ε-nets with respect to axis-parallel rectangles. They conjecture that $\varepsilon_k^{\square} \leq \frac{2}{k+3}$ ("there is always an $\frac{2}{k+3}$-net with k points"), which if true would mean this technique is also effective for high values of k. Dulieu [16] and Rachek [22] both show that $\varepsilon_4^{\square} \leq \frac{2}{7}$, together with the bounds by Aronov *et al.* confirming the conjecture for $k \leq 5$.

2.4 Personalised L_1 metric

We can also use the *personalised L_1 metric* as defined by De Berg *et al.* [8]. This models the fact that a voter v might assign more weight to certain topics. To that end, each voter v is given a personal preference vector $w(v) = (w_1(d), \ldots, w_d(v)) \in \mathbb{R}^d$. The distance between a voter $v \in V$ and a point $p \in \mathbb{R}^d$ is then defined as $\text{dist}_w(v,p) = \sum_{i=1}^d w_i(v)|p_i - v_i|$. Worth noting is that this is not actually a metric: it is only defined when the first argument is a voter and it is not necessarily symmetric.

To simplify things, we assume the preference vectors are normalised, i.e. $\|w(v)\| = 1$. This does not restrict the problem, because we only care which points a voter considers the closest and multiplying the preference vector by a scalar does not change that.

We only need to slightly alter Lemma 2.3.1 to make it work here:

Lemma 2.4.1. *Let $p \in \mathbb{R}^d$ be a point played by Player 1. For a vector $c \in \mathbb{R}^d$, define the set of voters*

$$V_c^p = \{v \in V \mid c_i(v_i - p_i) \geq 0 \text{ and } w_i(v) = |c_i| \text{ for all } 1 \leq i \leq d\}.$$

A point q by Player 2 that wins a voter $v \in V_c^p$ must have inner product $\langle c, q\rangle > \langle c, p\rangle$.

Proof. Let $v \in V_c^p$. For any i we know $c_i(v_i - p_i) \geq 0$ and $|c_i| = w_i(v)$, therefore $c_i(v_i - p_i) = w_i(v)|v_i - p_i|$ and

$$\text{dist}_w(v,p) = \sum_{i=1}^{d} w_i(v)|v_i - p_i| = \sum_{i=1}^{d} c_i(v_i - p_i) = \langle c, v - p\rangle.$$

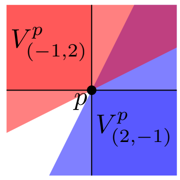
Because $|c_i| = w_i(v)$ for all i we also know that

$$\text{dist}_w(v,q) = \sum_{i=1}^{d} w_i(v)|v_i - q_i| \geq \sum_{i=1}^{d} c_i(v_i - q_i) = \langle c, v - q\rangle.$$

If we assume q wins v, then $\text{dist}_w(v,q) < \text{dist}_w(v,p)$. This now gives us

$$\langle c, v-q \rangle \leq \text{dist}_w(v,q) < \text{dist}_w(v,p) = \langle c, v-p \rangle,$$

which means $\langle c,q \rangle > \langle c,p \rangle$. □



Here, there still is no vector c where Player 2 can win voters from both V^p_c and V^p_{-c}. On the other hand, it is now possible to win voters from opposing orthants O^p_c and O^p_{-c}, provided those voters have different preferences. The figure on the right shows this: the voters $V^p_{(-1,2)}$ are those that have preference $(1,2)$ that are somewhere in the red orthant $O^p_{(-1,1)}$ and can be won if Player 2 places a point in the bigger red area, while the voters $V^p_{(2,-1)}$ have preference $(2,1)$, are in the blue orthant $O^p(1,-1)$ and can be won with a point in the bigger blue area. A point placed where the two areas overlap might thus win voters from the two opposing orthants.

We will show that Player 2 still cannot win voters in O^p_c when placing a point in O^p_{-c}. This lets us give strategies for Player 1 without having to worry about the specific preference vectors.

2.4.1 Preference-oblivious methods

As was the case for the L_2 and L_1 metric, we can use an ε-net to get a solution.

Theorem 2.4.2. *Let V be a set of n voters in $\mathbb{R}^d$. If Player 1 places points according to an ε-net for V with respect to axis-parallel star-shaped sets, then Player 2 always wins less than εn voters.*

Proof. Assume Player 1 placed a point p and Player 2's point q lies in orthant O^p_c while voter v lies in the opposite orthant O^p_{-c}. For any i, we know $c_i(q_i - p_i) = |q_i - p_i|$ and $-c_i(v_i - p_i) = |v_i - p_i|$. Thus, $|q_i - p_i| + |v_i - p_i| = c_i(q_i - v_i)$, which because $|c_i| = 1$ must be equal to $|q_i - v_i|$. Therefore, $|q_i - v_i| \geq |v_i - p_i|$. Regardless of the preference vector $w(v)$, we now have $\text{dist}_w(v,q) \geq \text{dist}_w(v,p)$ as well, which means q cannot win v. The voters won by q must thus lie in $\mathbb{R}^d \setminus O^p_{-c}$.

This means that for any Player 1 point p there is an axis-parallel set that is star-shaped with respect to q, covering all voters q wins without containing p. When we look at all points Player 1 placed, then we can intersect those sets to get a set S that is still axis-parallel, star-shaped with respect to q and covering all voters q wins. On top of that, it cannot contain any of Player 1's points. Assuming Player 1's points are placed according to an ε-net for V with respect to axis-parallel star-shaped sets, then that means the set S must cover less than εn voters. Since all voters won by q are covered by S, Player 2 must also win less than εn voters. □

As can be expected no research has been done on these specific ε-nets, so this subsection will establish bounds for it. Similarly to the ε-nets before this, we will use $\varepsilon^{\mathcal{S}}_k$ to denote the lowest value such that any point set has a k-point $\varepsilon^{\mathcal{S}}_k$ net.

One technique Aronov *et al.* [2] use to make ε-nets for axis-parallel rectangles is to place the points according to a grid where the rows and columns all have varying heights or widths to ensure that they contain the same number of voters. To avoid intersecting the ε-net, an axis-parallel rectangle can then only cover one column or one row, which means it can cover $\frac{n}{i+1}$ voters in an $i \times i$ grid.

In the same situation, an axis-parallel star-shaped set can cover both one column and one row. Thus, Player 2 can win at most $\frac{n}{i_x+1} + \frac{n}{i_y+1}$ voters when Player 1's points are placed according to a $i_x \times i_y$ grid. To improve the grid, Aronov *et al.* [2] construct ε-nets inside the rows and columns, which is something we can also use here. When we put r_x-point $\varepsilon^{\mathcal{S}}_{r_x}$-nets in j_x of the rows and r_y-point $\varepsilon^{\mathcal{S}}_{r_y}$-nets in j_y of the columns, we get the following bounds.

Lemma 2.4.3. *For all positive integers $i_x, i_y, j_x, j_y, r_x, r_y$ with $j_x \le i_x$ and $j_y \le i_y$,*

$$\varepsilon^{\mathcal{S}}_{i_x i_y + j_x r_x + j_y r_y} \le \frac{1}{j_x/\varepsilon^{\mathcal{S}}_{r_x} + i_x + 1 - j_x} + \frac{1}{j_y/\varepsilon^{\mathcal{S}}_{r_y} + i_y + 1 - j_y}.$$

For $k = 1$ this only gives the trivial bound $\varepsilon^{\mathcal{S}}_1 \le 1$, but as it turns out this bound is actually tight. We can see this when all voters lie on the line $y = x$. When Player 1 places a point p above or below the line, then $O^p_{(1,-1)}$ or $O^p_{(-1,1)}$ cannot contain voters. When p lies on the line, then neither $O^p_{(1,-1)}$ nor $O^p_{(-1,1)}$ can contain voters. Thus, regardless of where p is placed there is always an orthant without voters, and one of the star-shaped axis-parallel sets $\mathbb{R}^2 \setminus O^p_{(1,-1)}$ and $\mathbb{R}^2 \setminus O^p_{(-1,1)}$ will cover all voters without containing p.

For $k = 2$ we can do better. As Figure 2.3 shows, with the grid construction we always count the voters in the same cell as Player 2's point twice, but we can use a different construction to avoid that. There, we first take a horizontal line to divide the voter set in an upper and a lower half both containing half the voters, and then for both halves find the vertical line that divides it in half. Placing Player 1's points where those lines intersect then ensures that there is always $\frac{1}{4}$ of the voters that Player 2 cannot win.

We can make similar constructions with three points to guarantee Player 1 wins $\frac{5}{7}$ voters and five points for $\frac{7}{11}$. These are also shown in Figure 2.3. For higher numbers of points, these constructions are no longer better than the grid construction. One explanation for this is that the cell for which the voters get counted twice becomes small enough to not matter as much anymore.

Lemma 2.4.4. $\varepsilon^{\mathcal{S}}_2 \le \frac{3}{4}$ *and* $\varepsilon^{\mathcal{S}}_3 \le \frac{5}{7}$ *and* $\varepsilon^{\mathcal{S}}_5 \le \frac{7}{11}$.

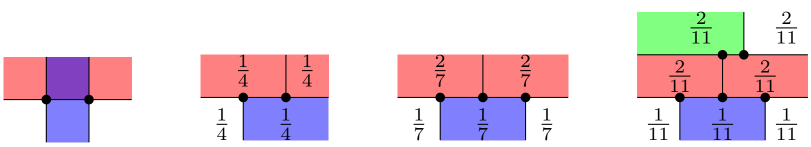


Figure 2.3: The grid construction with two points, and alternative constructions with two, three and five points

2.4.2 Preference-aware methods

In the previous subsection, we did not actually consider the preference vectors of the voters, but simply made sure the methods work regardless of what they are. As can be expected, taking the preferences into account gives better methods. However, how good they are depends on how many different preference vectors there are. To capture this, we let $W = \{w(v) \mid v \in V\}$ denote the set of used preference vectors. In the worst case we can have $|W| = n$, which as we will see means the preference-aware methods do not give useful results.

When we only have $|W| = 1$, we can re-scale the voter set to bring it back to the normal L_1 metric and then solve it as before. For placing a single point, this uses the medians of the x-coordinates and the y-coordinates in the voter set. We can do something similar with *weighted medians* when $|W| = 2$.

Definition 2.4.5. *Let $V \subset \mathbb{R}$ be a set where each $v \in V$ has weight $\omega(v) \in \mathbb{R}_{\geq 0}$. Then the value $x \in \mathbb{R}$ is a* weighted median *when*

$$\sum_{\substack{v \in V \\ v < x}} \omega(v) \leq \frac{1}{2} \sum_{v \in V} \omega(v) \quad \textit{and} \quad \sum_{\substack{v \in V \\ v > x}} \omega(v) \leq \frac{1}{2} \sum_{v \in V} \omega(v).$$

At least one weighted median always exists.

Lemma 2.4.6. *When $|W| = 2$, Player 1 can place one point to make sure Player 2 can win at most $\frac{n}{\sqrt{2}}$ voters.*

Proof. Assume $W = \{(w'_x, w'_y),\ (w''_x, w''_y)\}$ with $w'_x > w''_x$. The preference vectors are normalised, so also $w'_y < w''_y$. Let V' be the voters with preference (w'_x, w'_y) and V'' those with preference (w''_x, w''_y). We place the point $p \in \mathbb{R}^2$ based on weighted medians:

- Its x-coordinate is a weighted median for the voters' x-coordinates where a voter $v \in V''$ has weight $\omega(v) = \sqrt{2} - 1$ and the rest has weight 1.
- Its y-coordinate is a weighted median for the voters' y-coordinates where a voter $v \in V'$ has weight $\omega(v) = \sqrt{2} - 1$ and the rest has weight 1.

We can use a linear program to bound the number of voters Player 2 can win. As variables we use $v'_{++}, v'_{+-}, v'_{--}, v'_{-+}, v''_{++}, v''_{+-}, v''_{--}, v''_{-+} \in \mathbb{R}_{\geq 0}$, where for example v'_{+-} corresponds to $|V^p_{(+w'_x, -w'_y)}|/n$; the voters from V' in the bottom-right quadrant. We will also use v' as shorthand for $v'_{++} + v'_{+-} + v'_{--} + v'_{-+}$ and define v'' analogously.

As first constraint, we have $v' + v'' = 1$. The construction of p as a weighted median also brings constraints:

$$v'_{++} + v'_{+-} + (\sqrt{2}-1)\left(v''_{++} + v''_{+-}\right) \leq \frac{1}{2}\left(v' + (\sqrt{2}-1)v''\right)$$
$$v'_{-+} + v'_{--} + (\sqrt{2}-1)\left(v''_{-+} + v''_{--}\right) \leq \frac{1}{2}\left(v' + (\sqrt{2}-1)v''\right)$$
$$(\sqrt{2}-1)\left(v'_{++} + v'_{-+}\right) + v''_{++} + v''_{-+} \leq \frac{1}{2}\left((\sqrt{2}-1)v' + v''\right)$$
$$(\sqrt{2}-1)\left(v'_{+-} + v'_{--}\right) + v''_{+-} + v''_{--} \leq \frac{1}{2}\left((\sqrt{2}-1)v' + v''\right)$$

With the objective function we count how many of the voters Player 2 can win under these constraints. Figure 2.4 depicts which combinations of voters Player 2 can win. The blue lines belong to V' while the red lines belong to V'', in the sense that for example a point above both red lines might win all voters from V'' above p. Player 2 only wins voters on one side (above/below/left/right) of p when placing a point in a light grey area, while winning voters from V' to the left or right of p and from V'' above or below p when placing a point in a dark grey area.

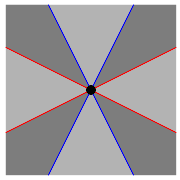


Figure 2.4: Wedges where Player 2's point wins a different combination of voters

Due to symmetry all light grey and all dark grey areas behave the same, so as objective we maximise the maximum of what Player 2 would win by placing a point in the rightmost area, and what a point in the top-right area would win:

$$\max\{v'_{++} + v'_{+-} + v''_{++} + v''_{+-}, \quad v'_{++} + v'_{+-} + v''_{++} + v''_{-+}\}$$

Solving this linear program gives $1/\sqrt{2}$ as highest attainable value. In other words, with any distribution of voters, placing Player 1's point as described ensures Player 2 wins at most $n/\sqrt{2}$ voters. $\square$

For $|W| \geq 3$ constructions with weighted medians also work, but there is no obvious pattern to which weights should be used and what result it gives for Player 2.

To get a general preference-aware solution, we can combine the method for $|W| = 1$ and the weighted-median method. For this, we let the voters for each of the m most common preferences be handled separately with the former method, the voters for k_0 pairs of the remaining most common preferences with the latter method, and leave any remaining voters as is. This gives us Lemma 2.4.7. For some combinations of $|W|$ and k, the best bounds given by the lemma or the preference-oblivious method are listed in Table 2.1.

Lemma 2.4.7. *By placing $k = k_0 + \sum_{i=1}^{m} k_i$ points, Player 1 can ensure the fraction of voters Player 2 wins is at most*

$$\frac{|W| - 2k_0 - m}{|W|} + \frac{1}{\sqrt{2}} \cdot \frac{2k_0}{|W|} + \sum_{i=1}^{m} \frac{\varepsilon^{\square}_{k_i}}{|W|}.$$

$\lvert W\rvert \setminus k$	1	2	3	4	5
1	0.5	0.4	0.333	0.286	0.25
2	0.707	0.5	0.45	0.4	0.367
3	0.805	0.638	0.5	0.467	0.433
4	0.854	0.707	0.604	0.5	0.475
5	0.883	0.75	0.666	0.646	0.5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
∞	1	0.75	0.714	0.667	0.636

Table 2.1: Upper bounds for how many voters Player 2 wins in the personalised L_1 metric

Chapter 3

Solutions for high k

When Player 1 is allowed to place many points (k is high), we get good strategies by using a compressed quadtree of the voter set to decide where to place points.

3.1 Algorithm

First, we construct a *compressed quadtree* $\mathcal{T}$ based on the voter set V. This can be done in $\mathcal{O}(n \log n)$ time and gives a tree structure where each node ν is associated with a square that includes its bottom and left edge but excludes the other two. Each internal node of this tree has four children. We denote the set of children of a node ν by $C(\nu)$.

In a compressed quadtree, we have at most $4n$ leaf nodes and for those the associated squares always contain at most one voter each. We want to more generally find a set $\mathcal{R}$ of regions such that the number of voters covered by a region is bounded by a number of our choosing, while keeping $|\mathcal{R}|$ low and maintaining a nice structure. To that end, we will go through the tree from the leaves up and only pick the regions that lead to more than m voters getting covered, for some given constant m.

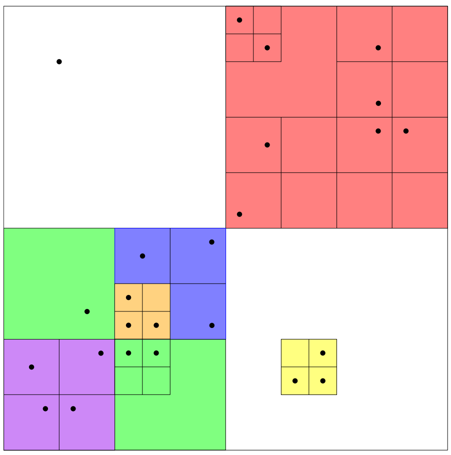


Figure 3.1: The compressed quadtree, and the region division for $m = 2$

Figure 3.1 shows the resulting regions for an arbitrary voter set with $m = 2$. The six regions each cover between three and eight voters, and one 'free' voter is left uncovered. The exact procedure is described by Algorithm 1.

Algorithm 1: MakeRegions(V, m)

Input: A quadtree node ν
Output: A set of regions $\mathcal{R}$ and a set of free voters V_{free}

Let $R(\nu)$ be the square corresponding to ν;
if ν *is a leaf node* **then**
  if ν *contains a voter* v **then**
    return $(\emptyset, \{v\})$
  else
    return $(\emptyset, \emptyset)$
  end
else
  Recursively call MakeRegions for all nodes in $C(\nu)$;
  Let $\mathcal{R}$ be the union of the returned sets of regions;
  Let V_{free} be the union of the sets of returned free voters;
  $R(\nu) \leftarrow R(\nu) \setminus \bigcup_{R \in \mathcal{R}} R$;
  if $|V_{\text{free}}| \leq m$ **then**
    return $(\mathcal{R}, V_{\text{free}})$
  else
    return $(\mathcal{R} \cup \{R(\nu)\}, \emptyset)$
  end
end

For the analysis it will often be useful to look at 'child' regions, so we define $C(R(\nu)) = \{R(\nu') \mid \nu' \in C(\nu))\}$ and we define the child regions of a collection $\mathcal{R}$ of regions to be $C(\mathcal{R}) = \bigcup_{R \in \mathcal{R}} C(R) \setminus \mathcal{R}$. That is, $C(\mathcal{R})$ contains the regions that are a child of a region in $\mathcal{R}$ but that are not in $\mathcal{R}$ themselves.

Both $\mathcal{R}$ and $C(\mathcal{R})$ have some useful properties. First of all, we can easily bound the number of regions and the number of voters in a region.

Lemma 3.1.1. *For any region $R \in C(\mathcal{R})$ we have $|V \cap R| \leq m < \frac{n}{|\mathcal{R}|}$.*

Proof. From the algorithm we get $R \in \mathcal{R}$ if and only if $|V \cap R| > m$. By definition $R \in C(\mathcal{R})$ means $R \notin \mathcal{R}$, so this shows the first part of the statement. Because the regions in $\mathcal{R}$ do not overlap and each covers more than m voters, the equivalence also means that $|\mathcal{R}| < \frac{n}{m}$. This can be rewritten to $m < \frac{n}{|\mathcal{R}|}$. □

Once we have these regions, we can use them to place points for Player 1. We will see that placing points on the corners of each region $R(\nu) \in C(\mathcal{R})$ ensures that Player 2 can only win voters from three different child regions (see Figure 3.2 for the simplest case). We also need the four points on the outside of each region $R(\nu) \in \mathcal{R}$ to ensure

Player 2 cannot win voters from multiple 'ancestor' regions; regions $R(\nu') \in \mathcal{R}$ where node ν' is an ancestor of ν. This gives the point set $P(R)$ as shown in Figure 3.2. Recall that k is the number of points placed by Player 1. Because we place 13 points per region $R(\nu) \in \mathcal{R}$, we have $k \leq 13|\mathcal{R}|$ and thus $m < 13\frac{n}{k}$.

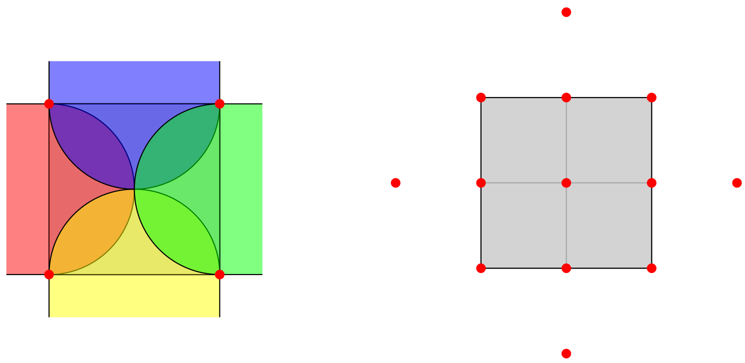


Figure 3.2: Left: If Player 2 places a point in a coloured semi-circle it can win voters in the region with that colour. Right: The 13 points $P(R)$ placed around a region $R \in \mathcal{R}$.

As noted, constructing a compressed quadtree takes $\mathcal{O}(n \log n)$ time and gives $\mathcal{O}(n)$ nodes. For each of the nodes, Algorithm 1 does a constant-time operation, and $|\mathcal{R}| = \mathcal{O}(n)$ so adding the points $P(R)$ for each region $R \in \mathcal{R}$ can also be done in $\mathcal{O}(n)$ time. Thus, the whole procedure has a running time of $\mathcal{O}(n \log n)$. The following lemma summarises the properties of the construction.

Lemma 3.1.2. *The quadtree-based method described above places less than $13\frac{n}{m}$ points of Player 1 and runs in $\mathcal{O}(n \log n)$ time.*

Next we analyse the number of voters that Player 2 can win.

3.2 A first bound

To give an upper bound for the number of voters Player 2 can win, we will bound the number of child regions Player 2 can win voters from. In the case depicted in Figure 3.2 it is not too complicated to prove Player 2 can only win voters from three child regions, but when we account for how a region can be nested in arbitrarily many ancestor regions this becomes more involved.

We can, however, already conclude a simple property from the fact that each child region has points on its corners. For a point q to win voters from a child region it needs to be closer to them than these corner points, which means it must be placed in the blue area in Figure 3.3. The blue area comes from the union of open disks that are centred around a possible voter in the child region and do not contain one of Player 1's points. We capture this idea with Definition 3.2.1:

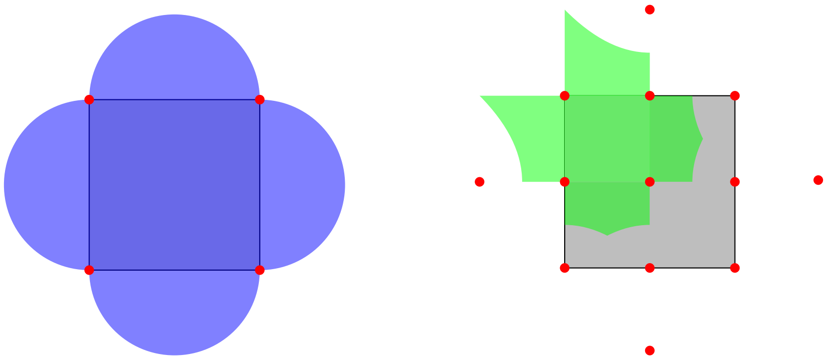


Figure 3.3: $Q(R)$ and $V(R)$ for any $R \in C(\mathcal{R})$

Definition 3.2.1. *For a region R, we define $Q(R)$ as the set of locations for Player 2's point that might win a voter in R, i.e.*

$$Q(R) = \{q \in \mathbb{R}^2 \mid \|q - v\| < \min_{p \in P} \|p - v\| \text{ for some } v \in R\}$$

We can also do the reverse and determine the area a point q can win voters in, when we know in which child region q is placed. This comes down to looking at the union of possible Voronoi cells, as shown in Figure 3.3. The boundary is formed by the parabolas that are equidistant from a point of Player 1 and a side of the child region.

Definition 3.2.2. *For a region R, we define $V(R)$ as the set of locations for voters that Player 2 might win by placing a point in R, i.e.*

$$V(R) = \{v \in \mathbb{R}^2 \mid \|q - v\| < \min_{p \in P} \|p - v\| \text{ for some } q \in R\}$$

From $C(\mathcal{R})$, let R_0 be the region containing q and R_l, R_r, R_u, R_d the first new regions encountered when moving from q in each of the cardinal directions (as depicted in Figure 3.4). Using the two properties we found, the following lemma will show that q can only win voters from these five regions of $C(\mathcal{R})$, which already gives a bound for how many voters Player 2 can win.

Lemma 3.2.3. *A point $q \in R_0$ can only win voters from R_0 and the first regions encountered in each of the cardinal directions (R_l, R_r, R_u and R_d).*

Proof. The point q can only win voters from regions $R \in C(\mathcal{R})$ that intersect the horizontal or vertical line through q, because otherwise it cannot lie in $Q(R)$. We follow the line from q in the given cardinal direction and let R be the first region of $C(\mathcal{R}) \setminus R_0$ encountered that we win voters from.

First, assume R is an ancestor of R_0. The quadtree structure means that the distance between R_0 and the outer boundary of R will be a multiple of the side length of R_0. The distance cannot be zero, because otherwise we cannot have encountered R. Thus the distance is at least the side length of R_0, meaning anything outside of R will definitely lie

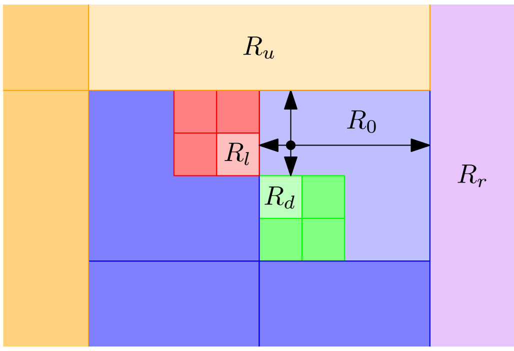


Figure 3.4: The five child regions Player 2's point might win voters from

outside of $V(R_0)$. Also because of the quadtree structure, the distance between R_0 and another quadtree region R' must be at least the minimum of their side lengths. Thus, when R' is at least as large as R_0 then it will be outside of $V(R_0)$, and R_0 itself will be outside of $Q(R')$ when R' is smaller than R_0. Thus, we can only win voters from R.

Now, assume R is not an ancestor. The points on the corners of R prevent us from winning any voters behind it. Hence, if q can win any voters from a region $R' \neq R$ that lies in the same cardinal direction as R, then R' must be a descendant of R. In that case, either R' touches the boundary of R (in which case we could not have won voters from R) or R' at least the side length of R' removed from the boundary (in which case q cannot lie in $Q(R')$). Thus, for a given cardinal direction we can always only win voters from the first encountered child region. □

3.3 A tight bound

From Lemma 3.2.3 we know we only need to consider five regions. However, it is not possible to get voters from all combinations of these regions. In fact Player 2 can only win voters from three, which gives a tighter bound.

To get this new bound, we will need to look at what happens when q is placed outside of a region but still wins voters from it. If we assume q is above the region, then it must be in the blue half-disk in Figure 3.5, and thus any voters won must be in the green area. The boundary for that is formed by the hyperbolas that are equidistant from the disk and Player 1's points (for the top and left) and the lines that are equidistant from the rightmost point of the disk and Player 1's points (for the bottom and right).

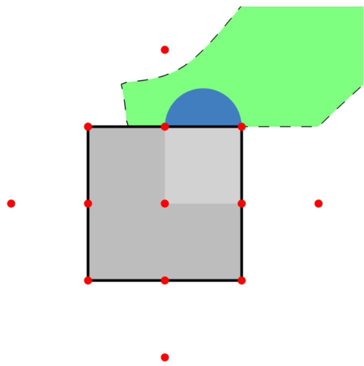


Figure 3.5: Player 2's point (in the blue area) can only win voters from the green area

We can now use this property to further limit the number of regions Player 2 can simultaneously win voters from.

Lemma 3.3.1. *A point $q \in R_0$ can only win voters from R_0 and at most two of the regions R_l, R_r, R_u, R_d.*

Proof. Let R be the smallest region in $\mathcal{R}$ the point q wins voters from. Without loss of generality assume R has side length 4 and its top-right corner is placed at the origin.

First, assume $q = (x, y)$ lies outside R. Without loss of generality we also assume that $y > 0$ and $x > -2$. For q to win voters from R we now also need $y < 2$ and $x < 0$. Since it already wins voters from R downwards, by Lemma 3.2.3 we cannot win voters from any other region below q. Because R is the smallest region, R_l cannot overlap the green area from Figure 3.5 either.

Thus, assume we win voters from both R_u and R_r. Let (u_x, u_y) and (r_x, r_y) denote the position of a won voter in R_u and R_r respectively. Without loss of generality, assume (u_x, u_y) lies on the lower boundary of R_u and (r_x, r_y) on the left boundary of R_r. Because R is the smallest region that q wins voters from and has side length 4, that means u_y and r_x should be multiples of 4. One of R_u and R_r will always have a corner point at (r_x, u_y), or two that lie closer to (u_x, u_y) and (r_x, r_y), thus the distance of (u_x, u_y) and (r_x, r_y) to their nearest corner point will be at most $r_x - u_x$ and $u_y - r_y$ respectively. Because we had $x < 0$ and $y < 2$, the distance to q is always at least $u_y - 2$ and r_x respectively. If q is to be closer we definitely need $u_y - 2 < r_x - u_x$, which means $r_x > u_x + u_y - 2$. We also need $r_x < u_y - r_y$, but that means r_x must be in the interval $(u_y + u_x - 2, u_y - r_y)$. As can be gathered from Figure 3.5, we definitely need $u_x \geq -2$ and $r_2 \geq 0$, so this will be at best $(u_y - 4, u_y)$. Because u_y is a multiple of 4 this interval cannot contain any others, which means there are no valid positions for r_x. Thus, we cannot win voters from both R_u and R_r.

Now, assume q lies inside R. Since R is the smallest region from which q wins voters, there are no child regions to account for. Thus, any other voters must come from outside of the quadrant q is in. In the worst case, these will be placed in the middle of the sides of the quadrant. Assuming the quadrant has side length 2, such a voter will be at most 1 unit away from the closest point of Player 1. Thus, if q is to win it, it should be less than 1 unit away from that side. This means q cannot win voters from two opposing sides, which means it can win from at most two cardinal directions. □

This now finally lets us conclude a good upper bound for the number of voters won by Player 2.

Theorem 3.3.2. *Let V be a set of n voters in $\mathbb{R}^2$. In $\mathcal{O}(n \log n)$ time, we can place k points for Player 1 such that by placing a single point Player 2 always wins less than $39\frac{n}{k}$ voters.*

Proof. From Lemma 3.3.1 we know that Player 2 can win voters from at most three regions of $C(\mathcal{R})$ and nowhere else. Lemma 3.1.1 gives us that these three regions each contain less than $\frac{n}{|\mathcal{R}|}$ voters. By construction $k \leq 13|\mathcal{R}|$, so finally Player 2 wins less than $3\frac{n}{|\mathcal{R}|} \leq 39\frac{n}{k}$ voters. As noted in Lemma 3.1.2, the construction takes $\mathcal{O}(n \log n)$ time. □

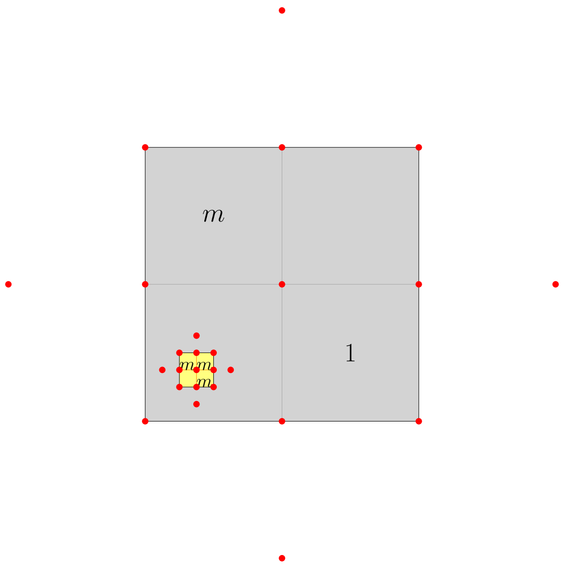


Figure 3.6: A setup that achieves the $39\frac{n}{k}$ bound

The setup from Figure 3.6 (where m means the region contains m voters) lets us get arbitrarily close to this bound, if we nest the grey region under a larger area the way the yellow region is nested under the grey region. This way, we have no overlaps and use $k = 13|\mathcal{R}|$ points for $n = |\mathcal{R}|(m+1) + 2m - 1$ voters. Thus, $13\frac{n}{k} = m + 1 + \frac{2m-1}{|\mathcal{R}|}$, which can get arbitrarily close to m if we make $|\mathcal{R}|$ and m sufficiently large. Player 2 can win $3m$ voters by placing a point in the top-right quadrant of the yellow region, thus this can get arbitrarily close to $39\frac{n}{k}$.

3.4 A more refined approach

Currently, we use the same 13 points regardless of where the voters lie in a region and how many there are, even though this can range from $m+1$ to $4m$. We can get better bounds by making the points depend on the number of voters and their locations.

One way to do this is to only place the full 13 points for regions that contain more than $\frac{3}{2}$ voters. For a region R with less voters, we save points by placing them only on the corners of R itself instead of the corners of all of its child regions. That gives the eight points shown in Figure 3.7. The region R now resembles what a child region looks like otherwise, and as can be seen $V(R)$ is still small.

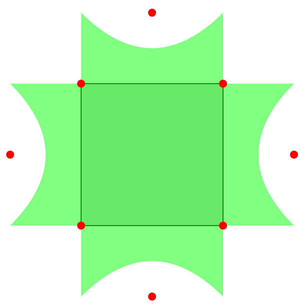


Figure 3.7: $V(R)$ for a region R where we leave out the innermost five points

Let $\mathcal{R}_1$ denote the regions with points on the corners of each child region and $\mathcal{R}_2$ those with points only on its own corners. Instead of $C(\mathcal{R})$, we will now look at $\tilde{C}(\mathcal{R}) = C(\mathcal{R}_1) \cup \mathcal{R}_2$. For each $R \in \tilde{C}(\mathcal{R})$ we know that there are points on the corners and that $Q(R)$ and $V(R)$ are small, so we can follow the exact same proof that lead to Lemma 3.3.1 to show that Player 2 can win voters from at most three regions in $\tilde{C}(\mathcal{R})$.

Now, a region in $\tilde{C}(\mathcal{R})$ can contain up to $\frac{3}{2}m$ voters. Despite that, we can ensure that Player 2 wins at most m voters from a region by placing centerpoints in the regions $\mathcal{R}_2$. The centerpoint is an $\frac{2}{3}$-net, and as such ensures that Player 2 win less than $\frac{2}{3} \cdot \frac{3}{2}m = m$ voters from a region in $\mathcal{R}_2$. Thus, when a region contains more than $\frac{3}{2}m$ voters we use nine points, while otherwise we use 13, and Player 2 still wins at most $3m$ voters in total.

When we include this case distinction in the algorithm, it lets us place less points to achieve the same result. Assuming $|\mathcal{R}_2| = x$ and $|\mathcal{R}_1| = y$, then the total number of points $k = 9x + 13y$ and also $x \cdot m + y \cdot \frac{3}{2}m < n$. Combining these two, $\frac{k-13y}{9}m + \frac{3}{2}ym = \frac{k+\frac{1}{2}y}{9}m < n$, which in the worst case means $m < 9\frac{n}{k}$. This is lower than the bound $m < 13\frac{n}{k}$ we got with the original setup, therefore means that the at most $3m$ voters Player 2 can win due to Theorem 3.3.2 only amount to $27\frac{n}{k}$ instead of $39\frac{n}{k}$.

We will need to find a centerpoint for $\mathcal{O}(|\mathcal{R}|)$ point sets, each containing $\mathcal{O}(m)$ points. Thus, this will take $\mathcal{O}(m \cdot |\mathcal{R}|)$ time [20], which is $\mathcal{O}(n)$ by Lemma 3.1.1. The running time of the full algorithm is therefore still $\mathcal{O}(n \log n)$.

3.4.1 Incorporating techniques for low k

For the previous construction, we used centerpoints to ensure Player 2 can win at most m voters from a region in $\tilde{C}(\mathcal{R})$, even though they might contain more. We can take this further and let Player 2 only win $\frac{2}{3}m$ voters from these regions. For a region in $\mathcal{R}$ that covers $4m$ voters, this is accomplished by placing the 13 points as before and then adding centerpoints for the voters in each quadrant. For regions containing less voters, we can leave out the centerpoint for any quadrant that contains at most $\frac{2}{3}m$ voters anyway. In the worst case we have i quadrants that contain $\frac{2}{3}m + 1$ voters each, requiring us to use $13 + i$ points for (slightly more than) $\frac{2}{3}m \cdot i$ voters. The number line in Figure 3.8 shows the maximum number of points a region needs using this technique when it contains a given number of voters.

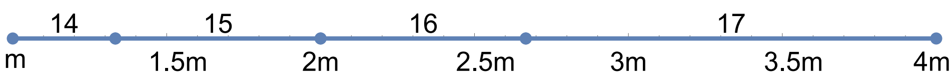


Figure 3.8: Maximum points needed for a region with a given number of voters

When a region contains only $m + 1$ voters, this still forces us to use 14 points for it. We can do better if, like before, we leave out the five innermost points. To make sure Player 2 can win at most $\frac{2}{3}$ voters, we now need to do more than place a centerpoint. Instead, we can place a two-point $\overline{\varepsilon_2^{\mathcal{C}}}$-net, as given in Chapter 2. For a region R, this works as long as $\overline{\varepsilon_2^{\mathcal{C}}} \cdot |R \cap V| \le \frac{2}{3}m$. As $\overline{\varepsilon_2^{\mathcal{C}}} = \frac{4}{7}$ this works when it contains at most $\frac{7}{6}m$

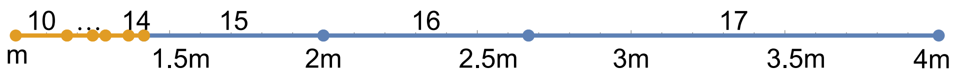


Figure 3.9: Maximum points needed for a region with a given number of voters

voters. In general, an $\overline{\varepsilon_i^{\mathcal{C}}}$-net lets us use $8+i$ points for up to $\frac{2}{3}m/\overline{\varepsilon_i^{\mathcal{C}}}$ voters.

We get the best results by using this new construction when placing up to 14 points, while otherwise still placing the 13 points and adding centerpoints to some quadrants. This gives the number line in Figure 3.9.

Table 3.1 gives the exact values involved in the case distinction: for every construction it gives the number of points needed and at which number of voters it starts being used. In all cases, this is when the previous construction stops working, e.g. the construction with 14 points is used when a region contains more than $1\frac{2}{5}m$ voters because the construction with 13 points can handle up to $\frac{2}{3}m/\overline{\varepsilon_5^{\mathcal{C}}} = 1\frac{2}{5}m$ voters.

In the last row, we divide the number of points used for a region by the minimum number of voters it covers. This gives a bound for m: if all regions use the same construction costing k' points and covering more than $n'm$ voters, then $k \leq |\mathcal{R}|k'$ and $n > |\mathcal{R}|n'm$ and therefore $m < \frac{k'}{n'} \cdot \frac{n}{k}$. When multiple constructions are used, this bound is somewhere between the values for only one construction. Here, we can see that in the worst case we only get $m < 10\frac{5}{16} \cdot \frac{n}{k}$. Player 2 can win at most $2m$ voters, so this amounts to less than $20\frac{5}{8} \cdot \frac{n}{k}$.

Number of points	10	11	12	13	14	15	16	17
Min. voters covered	m	$1\frac{1}{6}m$	$1\frac{1}{4}m$	$1\frac{1}{3}m$	$1\frac{2}{5}m$	$1\frac{5}{11}m$	$2m$	$2\frac{2}{3}m$
Max. points per m voters	10	$9\frac{3}{7}$	$9\frac{3}{5}$	$9\frac{3}{4}$	10	$10\frac{5}{16}$	8	$6\frac{3}{8}$

Table 3.1: Bounds resulting from the various setups

3.4.2 Excluding quadrants

The bound of $20\frac{5}{8}$ comes from the strategy with 14 points (of which six form an $\overline{\varepsilon_6^{\mathcal{C}}}$-net) only being able to cover up to $\frac{2}{3}m/\overline{\varepsilon_6^{\mathcal{C}}} = 1\frac{5}{11}m$ voters. The other strategy with 14 points is to place 13 of them as before, and the remaining one as centerpoint in the quadrant with the most voters. However, this already stops working when two quadrants have $\frac{2}{3}m+1$ voters each. If only those two quadrants have voters, we might as well leave out the points for the other quadrants, leading to one of the setups in Figure 3.10. With R and R' as the two remaining child regions, $Q(R), Q(R'), V(R), V(R')$ are all still small enough for Lemma 3.3.1 to hold.

Often, we will of course still have some voters in the other two quadrants. As long as these are at most m, we can simply pass these along as free voters in Algorithm 1 and let it put them in a region later.

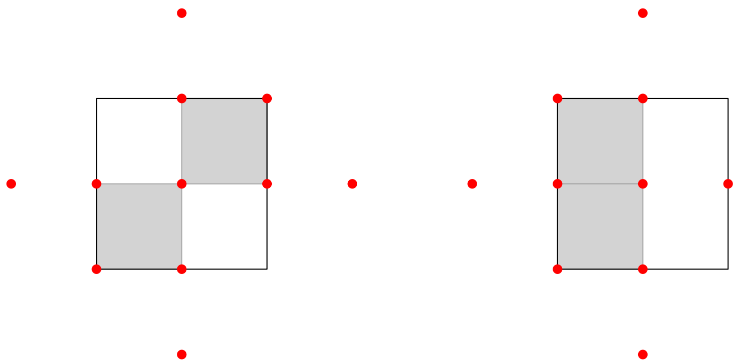


Figure 3.10: Possible setups when only two quadrants contain points

If we only use this setup when there are two quadrants with more than $\frac{2}{3}m$ voters and add centerpoints to both those quadrants, it means it always covers more than $1\frac{1}{3}m$ voters and can handle the region containing up to $2\frac{1}{3}m$. This gives us Table 3.2, where we can see that the new worst bound is $m < 10 \cdot \frac{n}{k}$, achieved when we are forced to use 10 or 14 points for every region. This means Player 2 always wins less than $20 \cdot \frac{n}{k}$ voters.

Theorem 3.4.1. *Let V be a set of n voters in $\mathbb{R}^2$. Player 1 can place k points to ensure that by placing a single point Player 2 wins less than $20\frac{n}{k}$ voters.*

Number of points	10	11	12	13	14	13	16	17
Min. voters covered	m	$1\frac{1}{6}m$	$1\frac{1}{4}m$	$1\frac{1}{3}m$	$1\frac{2}{5}m$	$1\frac{1}{3}m$	$2\frac{1}{3}m$	$2\frac{2}{3}m$
Max. points per m voters	10	$9\frac{3}{7}$	$9\frac{3}{5}$	$9\frac{3}{4}$	10	$9\frac{3}{4}$	$6\frac{6}{7}$	$6\frac{3}{8}$

Table 3.2: New bounds resulting from the various setups

3.5 L_1 metric

The quadtree method works well in the L_2 metric, but in the L_1 metric it works even better. We place the same 13 points as before based on the procedure in Section 3.1. As Figure 3.11 shows, in the simple situation where a region is a square, Player 2 can win outside voters from only one direction as opposed to the two we had under the L_2 metric. We will show that in more complicated situations this still holds.

Figure 3.11 also shows where a point placed in the blue triangle (such that it can win voters from the top-right quadrant) might win other voters. This is again much nicer than what we had in L_2, because there the equivalent area was unbounded. Here, it is bounded by the diagonal and horizontal line equidistant from the top of the triangle and Player 1's highest point, the vertical line equidistant from the left of the triangle and Player 1's point there, and the vertical line equidistant from the right of the triangle and Player 1's point there.

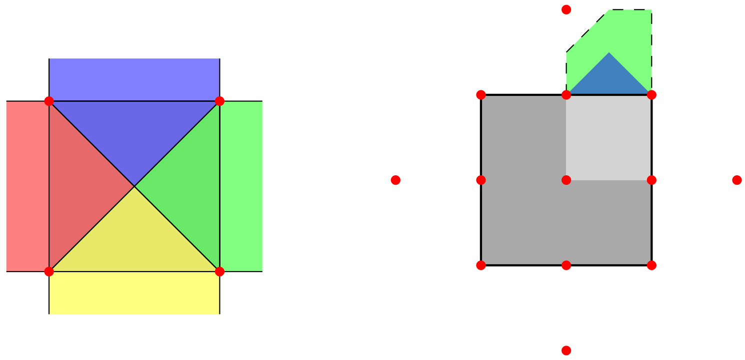


Figure 3.11: Left: If Player 2 places a point in a coloured triangle it can win voters in the region with that colour. Right: Player 2's point (in the blue area) can only win voters from the green area

Lemma 3.5.1. *q only wins voters from at most two regions of $C(\mathcal{R})$.*

Proof. Let R be the smallest region in $\mathcal{R}$ the point q wins voters from.

First, assume q lies inside R. Since R is the smallest region, there are no child regions to account for. That means the situation is as in Figure 3.11. Thus, in that case q can only win voters from at most two regions of $C(\mathcal{R})$.

Now assume q lies outside of R. As Figure 3.11 shows, there is only a small area where other voters won by q can lie. Because R is the smallest region q wins voters from, no other region can overlap that area. Therefore, q can again only win voters from two regions of $C(\mathcal{R})$.

□

This directly lets us conclude an upper bound on the number of voters won by Player 2 that is better than its equivalent in the L_2 metric.

Theorem 3.5.2. *Let V be a set of n voters in $\mathbb{R}^2$. In $\mathcal{O}(n \log n)$ time, we can find k points for Player 1 to place such that by placing a single point Player 2 always wins less than $26\frac{n}{k}$ voters.*

Proof. From Lemma 3.5.1 we know that Player 2 can win voters from at most two regions of $C(\mathcal{R})$ and nowhere else. Lemma 3.1.1 gives us that these two regions each contain less than $\frac{n}{|\mathcal{R}|}$ voters. By construction $k \leq 13|\mathcal{R}|$, so finally Player 2 wins less than $2\frac{n}{|\mathcal{R}|} \leq 26\frac{n}{k}$ voters. As noted in Lemma 3.1.2, this takes $\mathcal{O}(n \log n)$ time. □

3.5.1 A more refined approach

As with the L_2 metric, we can make a case distinction to further improve this result. Here, the ε-net solution from Chapter 2 is already very good (if as conjectured we can always make a $\frac{2}{k+3}$-net from k points, it is better to not use the quadtree technique at all). To make optimal use of that, we enforce that the opponent can only win $\frac{1}{6}m$ voters from any region in $C(\mathcal{R})$. Because $\overline{\varepsilon_{10}^{\square}} = \frac{1}{6}$ this means we place a staggering $13 + 4 \cdot 10 = 53$ points when a region covers $4m$ voters. However, this number is balanced out by the fact that the opponent can only win $2 \cdot \frac{1}{6}m = \frac{1}{3}m$ voters in total.

When placing $8+i \leq 27$ points we place the eight outer points and then the remaining i according to an $\overline{\varepsilon_i^{\square}}$-net. These can be used for up to $\frac{1}{6}m/\overline{\varepsilon_i^{\square}}$ voters. Table 3.3 shows the bounds this gives. The construction with 27 points works for up to $1\frac{5}{12}m$ voters.

Number of points	20	22	24	25	27
Min. voters covered	m	$1\frac{1}{12}m$	$1\frac{1}{6}m$	$1\frac{1}{4}m$	$1\frac{1}{3}m$
Max. points per m voters	20	$20\frac{4}{13}$	$20\frac{4}{7}$	20	$20\frac{1}{4}$

Table 3.3: Bounds resulting from the various setups

Otherwise, we place the full 13 points and use the remaining ones to make ε-nets in the child regions. For convenience, let $1/\overline{\varepsilon_{-1}^{\square}} = 0$. Now, a quadrant needs an $\overline{\varepsilon_i^{\square}}$-net when it contains more than $\frac{1}{6}m/\overline{\varepsilon_{i-1}^{\square}}$ voters. More generally, let $i_1, \ldots, i_4$ denote the number of points used for an ε-net in each of the quadrants. We need $\sum_{j=1}^{4} i_j$ points when each quadrant j contains more than $\frac{1}{6}m/\overline{\varepsilon_{i_j-1}^{\square}}$ voters. Thus, $13 + i$ points can be used for up to $\min\left\{\sum_{j=1}^{4} \frac{1}{6}m/\overline{\varepsilon_{i_j-1}^{\square}} \;\middle|\; \sum_{j=1}^{4} i_j = i - 1\right\}$ voters.

Table 3.4 shows the bounds this gives. Here, we skipped many possible setups, such as the one with 42 points. This is because limiting us to the shown cases does not give a worse overall bound but prevents the table from getting overly large.

Number of points	29	30	32	34	41	51	53
Min. voters covered	$1\frac{5}{12}m$	$1\frac{1}{2}m$	$1\frac{7}{12}m$	$1\frac{2}{3}m$	$2m$	$2\frac{1}{2}m$	$3\frac{1}{3}m$
Max. points per m voters	$20\frac{8}{17}$	20	$20\frac{4}{19}$	$20\frac{2}{5}$	$20\frac{1}{2}$	$20\frac{2}{5}$	$15\frac{9}{10}$

Table 3.4: Bounds resulting from the various setups

We always need at most $20\frac{4}{7}$ points per m voters, so the total number of points is $k < 20\frac{4}{7} \cdot \frac{n}{m}$. Thus, $m < 20\frac{4}{7} \cdot \frac{n}{k}$ and Player 2 can win $\frac{1}{3}m < 6\frac{6}{7} \cdot \frac{n}{k}$ voters.

Theorem 3.5.3. *Let V be a set of n voters in $\mathbb{R}^2$. Player 1 can place k points to ensure that by placing a single point Player 2 wins less than $6\frac{6}{7} \cdot \frac{n}{k}$ voters.*

Chapter 4

Conclusion

We have proven new bounds for the one-round discrete Voronoi game for the L_2 metric, L_1 metric and the personalised L_1 metric. For low k, these are based on ε-nets with a different range space depending on how the distance is calculated. Respectively, these are convex sets, axis-parallel rectangles and axis-parallel star-shaped sets. We have given a new construction for ε-nets for convex sets that for $k \geq 4$ gives results better than previously known. To our knowledge this is the first time ε-nets for axis-parallel star-shaped sets are considered, though we do draw heavily from existing constructions for axis-parallel rectangles.

For high k we get good results using quadtree-based methods for the L_2 and L_1 metric. Compressed quadtrees can be generated quickly so in the places we can find a running time this was a nice $\mathcal{O}(n \log n)$. However, for the best bounds we do not have algorithms, because ε-net constructions they use as a subroutine do not have an algorithm yet. Therefore we also do not have algorithms to achieve the best lower bounds.

It would have been nice to have a single lower bound for both low and high k, but it makes sense that the quadtree-based method does not provide one: it limits the number of regions Player 2 can win voters from, but when the method only creates few regions this is not very effective.

For the personalised L_1 metric we have not managed to find a method that works well for high k. The quadtree method does not work, for one because a voter v with preference $w(v) = (1, 0)$ does not care about y-coordinates and might thus be won by a point q placed in a far away region. As a consequence, we also do not know if Player 1 can win $\mathcal{O}\left(\frac{n}{k}\right)$ as in the L_2 and L_1 metric.

4.1 Future work

There are still many options for and generalisations of the one-round discrete Voronoi game that have not been researched but could be interesting.

4.1.1 Higher dimensions

This thesis only gave bounds and algorithms for the one-round discrete Voronoi game in $\mathbb{R}^2$, so a natural future step would be to try to generalise the results to higher dimensions.

For the quadtree-based methods, this would mean using octrees and higher dimensional 'hyperoctrees'. The main hurdle is to figure out how many points each region should have and where. Simply placing points at each corner of a child region means that in high enough dimensions it becomes possible to place a point on one side of the child region and win voters on the other side of it, which was impossible in $\mathbb{R}^2$ and one of the reasons it gave a good bound.

The grid-based ε-net constructions for axis-parallel rectangles and star-shaped sets easily generalise to $\mathbb{R}^d$: instead of placing the ε-net according to a $i_x \times i_y$ grid we place it as an $i_1 \times \cdots \times i_d$ grid. This should then ensure that any axis-parallel hyperrectangle overlaps less than $\max_{1 \leq j \leq d} \frac{1}{i_j+1}$ of the voters and any axis-parallel star-shaped set less than $\sum_{j=1}^{d} \frac{n}{i_j+1}$, unless they intersect the ε-net. From this we directly get bounds for the personalised L_1 metric for $\mathbb{R}^d$ in general. However, as explained in Section 2.3 we cannot simply use ε-nets for axis-parallel hyperrectangles for the L_1 metric. Since the L_1 metric is a special case of the personalised L_1 metric, we can use ε-nets for axis-parallel star-shaped sets, but it would be interesting to see if there is a technique that works better.

The ε-net construction for convex sets given in Section 2.2 might be difficult to generalise to higher dimensions. It is based on an equipartition of the voter set by hyperplanes that intersect in one common point, but for higher dimensions no obvious option has presented itself.

4.1.2 VG(k, ℓ) with $\ell > 1$

Another natural next step is to consider the game when Player 2 can place more than one point. Player 1 can of course always use the same strategy as for VG$(k, 1)$, so that Player 2 wins at most ℓ times as many voters as in VG$(k, 1)$. When k is much larger than ℓ, this should be the best strategy, because any good strategy for Player 1 will leave multiple disjoint options for Player 2 to win the same number of voters. When k and ℓ are closer together, it might be possible to force the subsets of voters two Player 2 points would win to have overlap. A simple but not very meaningful example is when we place a centerpoint: there are multiple options for Player 2 to win $\frac{2}{3}n$ voters with one point, but for any two points there must be overlap in the $\frac{2}{3}n$ voters they win, because it is impossible to win more than n voters.

4.1.3 Algorithms to achieve the found bounds

Chapter 2 gives constructions for ε-nets, but no actual algorithms to (efficiently) do those constructions. For most constructions it should not be too complicated to make an algorithm, but turning the proof from Theorem 2.2.1 for partitioning a point set with three concurrent lines into an algorithm might be less easy. Here, it might be possible to alter the $\mathcal{O}\left(n^2 \log n\right)$ algorithm that was found for an equipartition with those same three concurrent lines [7].

4.1.4 β-plurality

As mentioned in the introduction, β-plurality points have been introduced as an alternative to normal plurality points. Existing research focuses on VG$(1,1)$, but the concept can easily be generalised to VG$(k,1)$. There, the question is how how many of the voters Player 1 can win with a given β and k.

In general, we can say that it is not possible to always win more than $\frac{k}{k+1}$ of the voters. This can be seen when there are $n = k+1$ voters that are each at different positions: Player 2 can always place a point on top of a voter that Player 1 did not place a point on top of. Regardless of which distance function is used, this ensures Player 2 wins that voter, thereby leaving only the remaining k for Player 1.

For the L_2 metric, we can already see some interesting results when $\beta \leq \sqrt{d}$. Then, Player 1 can place points as if working in the L_1 metric and win at least as many voters. This comes from the fact that $\|x\|_2 \leq \|x\|_1 \leq \sqrt{d}\|x\|_2$ for any $x \in \mathbb{R}^d$. It means that when we have a voter v and two points p, q such that $\|v-p\|_1 < \|v-q\|_1$,

$$\beta\|v-p\|_2 \leq \beta\|v-p\|_1 < \beta\|v-q\|_1 \leq \|v-q\|_2.$$

More difficult problems arise when we try to find, for example, how big β can be while still ensuring that Player 1 wins VG$(2,1)$ under optimal play.

Apart from this, it might also be interesting to see how β-plurality works for the personalised L_1 metric. Filtser and Filtser [17] have found a lower bound $\sqrt{2}-1$ for the value of β needed to ensure a β-pluralty point exists in any metric space, but the personalised L_1 metric is not a metric and it does not seem like their proof can be adapted to work for it. However, when all preference vectors are of the form $(1, w)$ with $\beta \leq w \leq \frac{1}{\beta}$, Player 1 can again treat the game as if it is in the normal L_1 metric. This is because when for some voter v and points p, q we have $\|v-p\|_1 \leq \|v-q\|_1$ and $w(v) = (1, w)$ with $\beta \leq w \leq 1$, then also

$$\beta \operatorname{dist}_w(v,p) \leq w\|v-p\|_1 \leq w\|v-q\|_1 \leq \operatorname{dist}_w(v,q).$$

Simultaneously, when $\frac{1}{\beta} \geq w \geq 1$,

$$\beta \operatorname{dist}_w(v,p) \leq \beta w\|v-p\|_1 \leq \beta w\|v-q\|_1 \leq \operatorname{dist}_w(v,q).$$

When the preference vectors are far apart this can force β to be arbitrarily small, so it would be interesting to see if there is also a bound that does not depend on the specific preferences.

4.1.5 Measures

All shown constructions will likely also work when the voters are given by a continuous measure instead of a point set, or even as long as any atom of the measure has measure at most one. The ε-nets and proofs should generalise to these measures without much extra work. To construct a quadtree for the measure, we subdivide any square that has measure more than one until there are none left.

The Voronoi game with measures generalises both the discrete and the continuous Voronoi games and could for example apply when voters are represented by a region instead of a point. It could also be useful when the exact position of (some) voters is unknown and instead given by a probability distribution, though in that case winning two voters with probability $\frac{1}{2}$ counts for the same as winning one voter with probability 1. The algorithms and bounds would only consider the expected number of voters won, while the probability of Player 1 winning might be more suitable. For example, if there are two voters in total then winning two voters with probability $\frac{1}{2}$ means losing half the time, while always winning a single voter means Player 1 always wins. This could be another possibility for future work.

Bibliography

[1] Hee-Kap Ahn, Siu-Wing Cheng, Otfried Cheong, Mordecai J. Golin, and René van Oostrum. Competitive facility location along a highway. In Jie Wang, editor, *Computing and Combinatorics, 7th Annual International Conference, COCOON 2001, Guilin, China, August 20-23, 2001, Proceedings*, volume 2108 of *Lecture Notes in Computer Science*, pages 237–246. Springer, 2001.

[2] Boris Aronov, Franz Aurenhammer, Ferran Hurtado, Stefan Langerman, David Rappaport, Carlos Seara, and Shakhar Smorodinsky. Small weak epsilon-nets. *Comput. Geom.*, 42(5):455–462, 2009.

[3] Boris Aronov, Mark de Berg, Joachim Gudmundsson, and Michael Horton. On beta-plurality points in spatial voting games. *CoRR*, abs/2003.07513, 2020.

[4] Aritra Banik, Bhaswar B. Bhattacharya, and Sandip Das. Optimal strategies for the one-round discrete voronoi game on a line. *J. Comb. Optim.*, 26(4):655–669, 2013.

[5] Aritra Banik, Bhaswar B. Bhattacharya, Sandip Das, and Satyaki Mukherjee. The discrete voronoi game in r^2. *Comput. Geom.*, 63:53–62, 2017.

[6] Aritra Banik, Jean-Lou De Carufel, Anil Maheshwari, and Michiel H. M. Smid. Discrete voronoi games and ε-nets, in two and three dimensions. *Comput. Geom.*, 55:41–58, 2016.

[7] Astrid Belder, Mark Bouwman, and Gábor Damásdi. An algorithm for six-way equipartition. Found at `https://damasdigabor.web.elte.hu/sixway.pdf`.

[8] Mark de Berg, Joachim Gudmundsson, and Mehran Mehr. Faster algorithms for computing plurality points. *ACM Trans. Algorithms*, 14(3):36:1–36:23, 2018.

[9] Mark de Berg, Sándor Kisfaludi-Bak, and Mehran Mehr. On one-round discrete voronoi games. In Pinyan Lu and Guochuan Zhang, editors, *30th International Symposium on Algorithms and Computation, ISAAC 2019, December 8-11, 2019, Shanghai University of Finance and Economics, Shanghai, China*, volume 149 of *LIPIcs*, pages 37:1–37:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[10] Manuel Blum, Robert W. Floyd, Vaughan R. Pratt, Ronald L. Rivest, and Robert Endre Tarjan. Time bounds for selection. *J. Comput. Syst. Sci.*, 7(4):448–461, 1973.

[11] Boris Bukh. A point in many triangles. *Electron. J. Comb.*, 13(1), 2006.

[12] Thomas Byrne, Sándor P. Fekete, Jörg Kalcsics, and Linda Kleist. Competitive location problems: Balanced facility location and the one-round manhattan voronoi game. In Ryuhei Uehara, Seok-Hee Hong, and Subhas C. Nandy, editors, *WALCOM: Algorithms and Computation - 15th International Conference and Workshops, WALCOM 2021, Yangon, Myanmar, February 28 - March 2, 2021, Proceedings*, volume 12635 of *Lecture Notes in Computer Science*, pages 103–115. Springer, 2021.

[13] Timothy M. Chan. An optimal randomized algorithm for maximum tukey depth. In J. Ian Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 430–436. SIAM, 2004.

[14] Shuchi Chawla, Uday Rajan, R. Ravi, and Amitabh Sinha. Min-max payoffs in a two-player location game. *Oper. Res. Lett.*, 34(5):499–507, 2006.

[15] Otfried Cheong, Sariel Har-Peled, Nathan Linial, and Jirí Matousek. The one-round voronoi game. *Discret. Comput. Geom.*, 31(1):125–138, 2004.

[16] Muriel Dulieu. ε-nets faibles. Master's thesis, Université Libre de Bruxelles, June 2006.

[17] Arnold Filtser and Omrit Filtser. Condorcet relaxation in spatial voting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 5407–5414. AAAI Press, 2021.

[18] David Haussler and Emo Welzl. epsilon-nets and simplex range queries. *Discret. Comput. Geom.*, 2:127–151, 1987.

[19] Niek J. A. van Hulzen. Experimental analysis of one-round discrete voronoi games. Master's thesis, Eindhoven University of Technology, August 2019.

[20] Shreesh Jadhav and Asish Mukhopadhyay. Computing a centerpoint of a finite planar set of points in linear time. *Discret. Comput. Geom.*, 12:291–312, 1994.

[21] Nabil H. Mustafa and Saurabh Ray. An optimal extension of the centerpoint theorem. *Comput. Geom.*, 42(6-7):505–510, 2009.

[22] Vladyslav Rachek. Small weak epsilon-nets in families of rectangles. Bachelor's thesis, Jagiellonian University, August 2020.