

PLURALITYVETO: A Simple Voting Rule Achieving Optimal Metric Distortion

Fatih Erdem Kizilkaya* and David Kempe

University of Southern California

{FatihErdemKizilkaya, David.M.Kempe}@gmail.com

Abstract

The metric distortion framework posits that n voters and m candidates are jointly embedded in a metric space such that voters rank candidates that are closer to them higher. A voting rule’s purpose is to pick a candidate with minimum total distance to the voters, given only the rankings, but not the actual distances. As a result, in the worst case, each deterministic rule picks a candidate whose total distance is at least three times larger than that of an optimal one, i.e., has distortion at least 3. A recent breakthrough result showed that achieving this bound of 3 is possible; however, the proof is non-constructive, and the voting rule itself is a complicated exhaustive search.

Our main result is an extremely simple voting rule, called PLURALITYVETO, which achieves the same optimal distortion of 3. Each candidate starts with a score equal to his number of first-place votes. These scores are then gradually decreased via an n -round veto process in which a candidate drops out when his score reaches zero. One after the other, voters decrement the score of their bottom choice among the standing candidates, and the last standing candidate wins. We give a one-paragraph proof that this voting rule achieves distortion 3. This rule is also immensely practical, and it only makes two queries to each voter, so it has low communication overhead. We also show that a straightforward extension can be used to give a constructive proof of the more general Ranking-Matching Lemma of Gkatzelis et al.

We also generalize PLURALITYVETO into a class of randomized voting rules in the following way: PLURALITYVETO is run only for $k < n$ rounds; then, a candidate is chosen with probability proportional to his residual score. This general rule interpolates between RANDOMDICTATORSHIP (for $k = 0$) and PLURALITYVETO (for $k = n - 1$), and k controls the variance of the output. We show that for all k , this rule has expected distortion at most 3.

1 Introduction

Voting is a fundamental process for reaching consensus and plays a vital role in democracies, organizations and businesses. Even honeybees use a type of voting to decide among potential nest sites; each bee casts a *numerical* vote indicating the intensity of its preferences, and the site with the highest score wins [Seeley, 2010]. Quantifying preference intensity is not so easy for complicated problems that humans face; thus, the predominant approach is to elicit, from each voter, a preference *ranking* over candidates. This creates a need for a *voting rule* that determines the winner from these preferences. Numerous rules have been proposed over the years, with no consensus on a “best” rule.

A key contribution of computer science in this regard has been viewing the ranking of a voter as a proxy for the latent cost she¹ incurs if some candidate wins. Then, an optimal candidate can be defined as one minimizing the total cost, and vote aggregation can be interpreted as an optimization problem with missing information. Due to the missing information, a voting rule can be thought of as an approximation algorithm, whose worst-case performance is referred to as its *distortion* in this setting.

Without any structures on the costs, not much can be achieved in terms of distortion [Caragiannis *et al.*, 2017]. An important contribution due to Anshelevich *et al.* [2015] (see also the journal version [Anshelevich *et al.*, 2018] and recent surveys [Anshelevich *et al.*, 2021b; Anshelevich *et al.*, 2021a]) was to assume that the n voters and m candidates are jointly embedded in a metric space, and voters rank candidates by increasing distance.² This viewpoint is motivated by observing that each candidate exhibits a standpoint on various issues that voters care about, and each voter also has a standpoint on these issues that is presumably reflected on her ballot. One would then expect that voters rank candidates whose standpoints are “closer” to theirs higher. Note that voting rules do not have access to this space; the only available information is the rankings, which serve as an *ordinal* proxy for the (cardinal) distances. The worst-case approximation specifically for metric costs is called *metric distortion*. (For-

¹For ease of presentation, we use female pronouns for voters and male pronouns for candidates throughout.

²This assumption generalizes the classic notion of *single-peaked preferences* [Black, 1948; Moulin, 1980].

*Contact Author

mal definitions of all concepts are given in Section 2.)

The metric distortion framework has proved to be a fruitful analysis tool. In their initial work, Anshelevich *et al.* [2015] established a lower bound of 3 on the distortion of any deterministic voting rule, and showed that the Copeland rule nearly matches the lower bound by achieving distortion 5. Several subsequent papers worked towards closing this gap. Initially, the Ranked Pairs rule was conjectured to achieve distortion 3. This was disproved by Goel *et al.* [2017] who gave a lower bound of 5; Kempe [2020a] strengthened the lower bound to $\Omega(\sqrt{m})$. The first improvement over the upper bound of 5 was due to Munagala and Wang [2019], who achieved distortion $2 + \sqrt{5} \approx 4.23$ using a novel asymmetric variant of the Copeland rule. The distortion-3 conjecture was recently resolved in a breakthrough result by Gkatzelis *et al.* [2020], using a novel voting rule called PLURALITYMATCHING.

One of the main drawbacks of PLURALITYMATCHING is that it is unusually complex for a voting rule in the conventional sense. The winner is selected based on perfect matchings in certain bipartite graphs, which we will discuss shortly. Due to the complex nature of the voting rule, it is not even obvious that there always exists a winner in PLURALITYMATCHING; indeed, this existence proof was the main contribution of Gkatzelis *et al.* [2020] over the prior work of Munagala and Wang [2019] and Kempe [2020a]. The rule is also almost certainly too technical to be understood by the general public.

Our main contribution is an extremely simple voting rule, called PLURALITYVETO, which achieves the same optimal metric distortion of 3.

Under PLURALITYVETO, each candidate starts with a score equal to his *plurality score*, i.e., the number of first-place votes he receives. These scores are then gradually decreased; when the score of a candidate reaches zero, he is eliminated. Voters are processed one by one in an arbitrary order: when it is the turn of a voter, she decrements the score of her bottom choice among uneliminated candidates. Since the initial scores add up to the number of voters, all of the candidates will be eliminated at the end. The last eliminated candidate wins. Notice that this rule does not even require access to the voters’ full rankings. Aside from the top choices, it only requires from each voter her bottom choice among uneliminated candidates. Thus, when implemented via sequential queries to users, PLURALITYVETO also has low communication overhead. We summarize our main result as follows. (A formal presentation and proof are given in Section 3.)

Contribution 1. *PLURALITYVETO has the optimal metric distortion of 3 and can be implemented to require each voter to communicate only $O(\log m)$ bits to the voting rule.*

A candidate with a strict majority of first-place votes wins under PLURALITYVETO, regardless of the order in which voters are processed. Thus, one can think of the elimination process (or, as we call it, *multi-round veto*) as a runoff stage. This makes our rule conceptually simple as well; it is just plurality voting followed by multi-round veto. In this respect, it resembles instant runoff voting, which is used in national elections in several countries. As in instant runoff voting, PLURALITYVETO can be used by eliciting from each voter

her full ranking, so that the runoff stage can be run instantly. Alternatively, it can be arranged as a two-stage election in which voters first cast a vote for their top choice, and in the following stage, each voter cancels the vote of another voter.

PLURALITYMATCHING and its analysis are based on *domination graphs*, a family of bipartite graphs $G(c)$ (one per candidate c) between voters and voters defined in Section 2. Munagala and Wang [2019] and Kempe [2020a] had shown that if $G(c)$ has a perfect matching, then c has distortion at most 3. The key contribution of Gkatzelis *et al.* [2020] was to show that such a candidate c always exists. In fact, they proved a generalization, called the Ranking-Matching Lemma, which shows the existence of a graph with a *weighted* perfect bipartite matching in a more general class of bipartite graphs. We also give (in Section 4) a constructive one-paragraph proof of this more general Ranking-Matching Lemma, using a variant FRACTIONALVETO OF PLURALITYVETO which decreases weights fractionally, rather than integrally, in each step.

Finally, we generalize PLURALITYVETO to a class of randomized voting rules that choose a candidate with probability proportional to his residual score at the k^{th} round of PLURALITYVETO, which we refer to as k -ROUNDPLURALITYVETO. When $k = 0$, this more general rule specializes to the well-known rule RANDOMDICTATORSHIP, which chooses the top choice of a uniformly random voter. Hence, randomizing the outcome proportional to the scores achieves distortion $3 - 2/n$ when $k = 0$, as shown in [Anshelevich and Postl, 2016]. In Section 5, we show that k -ROUNDPLURALITYVETO in fact achieves distortion at most 3 for *all* k . In that way, it interpolates between RANDOMDICTATORSHIP (for $k = 0$) and PLURALITYVETO (for $k = n - 1$), and the parameter k controls the variance of the output, but does not affect the distortion. The intuition behind this is that PLURALITYVETO repeatedly decreases scores for “extreme” candidates, and thus is likely to end up with central ones. We elaborate on this intuition in the context of Peer Selection in Section 6.

Contribution 2. *k -ROUNDPLURALITYVETO has expected metric distortion at most 3 for any number of rounds k .*

This upper bound is almost tight, since any rule that can only elect candidates who are the top choice of at least one voter must have distortion at least $3 - o(1)$ [Gkatzelis *et al.*, 2020]. Also, since k controls the variance of the output, at a high level, this result relates to the work of Fain *et al.* [2019] who are not only interested in the expected distortion of rules, but also in the expected *squared* distortion, essentially forcing randomized rules to have lower variance in their distortion.

Other Related Work. The utilitarian analysis of voting rules through the lens of approximation algorithms was first suggested in [Boutilier *et al.*, 2015; Caragiannis and Procaccia, 2011; Procaccia, 2010; Procaccia and Rosenschein, 2006]. Boutilier and Rosenschein; Anshelevich *et al.* [2016; 2015] were the first to clearly articulate the tension between the objective of maximizing utility (or minimizing cost) and the available information, which is only ordinal; they also termed the resulting gap *distortion*. In the earlier work, such as [Boutilier *et al.*, 2015; Caragiannis and Procaccia, 2011; Procaccia, 2010; Procaccia and Rosenschein, 2006], the fo-

cus was on (positive) utilities, and no additional assumptions (such as metric costs) were placed on the utilities.

Recall that PLURALITYVETO only requires very limited communication from each voter, albeit in an n -round sequential algorithm. In this way, our work relates generally to studies of the tradeoff between communication and metric distortion. The recent papers [Fain *et al.*, 2019; Kempe, 2020b] establish related lower bounds: Fain *et al.* [2019] show that any voting rule that only obtains the top $k = O(1)$ candidates of each voter must have squared distortion $\Omega(m)$, in particular implying a bound of $\Omega(m)$ for the distortion of deterministic rules. Kempe [2020b] proves a lower bound of $\Omega(m/k)$ on the distortion of any deterministic voting rule that only obtains the candidates ranked by each voter in a set K of size $k = |K|$ of positions. Our voting rule avoids these lower bounds by obtaining the bottom candidate from a specified set for each voter; thus, for different voters, the candidates in different positions are queried.

Using randomization, communication can be drastically reduced even compared to our voting rule. Fain *et al.* [2019] present a RANDOMREFEREE mechanism: the mechanism asks two randomly chosen voters for their top choices, and then has a third voter choose between the two proposed candidates. This mechanism, which only requires access to the top choices of two voters plus one bit, achieves not only constant expected distortion, but constant expected *squared* distortion.

2 Preliminaries

Throughout, we use bold face for vectors, and denote the i^{th} component of a vector \mathbf{x} by x_i . Given a set S , let $\Delta(S)$ denote the probability simplex over S , i.e., the set of non-negative weight vectors over S that add up to 1.

An *election* is a tuple $\mathcal{E} = (V, C, \succ_V)$ consisting of a set of n voters V , a set of m candidates C and a *ranked-choice profile* $\succ_V = (\succ_v)_{v \in V}$; here, \succ_v is the *ranking* of voter v , i.e., a total order over C . We say that voter v ranks candidate c *higher* than candidate c' if $c \succ_v c'$. We also use $c \succsim_v c'$ when $c \succ_v c'$ or $c = c'$, in which case we say that v ranks c *weakly higher* than c' .

A *voting rule* f is an algorithm that returns a candidate $f(\succ_V) \in C$ given a ranked-choice profile \succ_V . We refer to $f(\succ_V)$ as the *winner* of the election \mathcal{E} using the voting rule f , or just as the winner of f if \mathcal{E} is clear from the context. For the most part, we will consider deterministic voting rules; in Section 5, we will also study randomized voting rules.

We refer to the candidate ranked highest by voter v as the *top choice* of v , and denote him by $\text{top}(v)$. The candidate ranked lowest by voter v is likewise referred to as the *bottom choice* of v . We use $\text{plu}(c)$ to denote the *plurality score* of candidate c , i.e., the number of voters whose top choice is c .

2.1 Metric Distortion

A *metric* over a set S is a function $d : S \times S \rightarrow \mathbb{R}_{\geq 0}$ which satisfies the following conditions for all $a, b, c \in S$: (1) Positive Definiteness: $d(a, b) = 0$ if and only³ if $a = b$;

³Our proofs do not require the “only if” condition, so technically, all our results hold for pseudo-metrics, not just metrics.

(2) Symmetry: $d(a, b) = d(b, a)$; (3) Triangle inequality: $d(a, b) + d(b, c) \geq d(a, c)$.

Given an election $\mathcal{E} = (V, C, \succ_V)$, we say that a metric d over⁴ $V \cup C$ is *consistent* with the ranking \succ_v of voter v if $d(v, c) \leq d(v, c')$ for all $c, c' \in C$ such that $c \succ_v c'$. We say that d is consistent with the ranked-choice profile \succ_V if it is consistent with the ranking \succ_v for all voters $v \in V$. We use $\mathcal{D}(\succ_V)$ to denote the domain of metrics consistent with \succ_V .

The (*utilitarian*) *social cost* of a candidate c with respect to a metric d is defined as the candidate’s sum of distances to all voters: $\text{cost}(c, d) = \sum_{v \in V} d(v, c)$. A candidate c_d^* is *optimal* with respect to the metric d if $c_d^* \in \arg \min_{c \in C} \text{cost}(c, d)$. The *distortion* of a voting rule f , denoted by $\text{dist}(f)$, is the largest possible ratio between the cost of the winner of f and that of an optimal candidate c_d^* , with respect to the worst possible metric $d \in \mathcal{D}(\succ_V)$. That is,

$$\text{dist}(f) = \max_{\succ_V} \sup_{d \in \mathcal{D}(\succ_V)} \frac{\text{cost}(f(\succ_V), d)}{\text{cost}(c_d^*, d)}.$$

2.2 Domination Graphs

Domination graphs offer a conceptually simple approach for giving an upper bound of 3 on the distortion of a voting rule. Given an election $\mathcal{E} = (V, C, \succ_V)$, the *domination graph* of a candidate $c \in C$ is the bipartite graph $G^{\mathcal{E}}(c) = (V, V, E_c)$ where $(v, v') \in E_c$ if and only if $c \succsim_v \text{top}(v')$. The main use of these graphs is via the following lemma, due to [Munagala and Wang, 2019; Kempe, 2020a; Gkatzelis *et al.*, 2020].

Lemma 1. *Let f be a voting rule such that for every election $\mathcal{E} = (V, C, \succ_V)$, the domination graph $G^{\mathcal{E}}(f(\succ_V))$ has a perfect matching. Then, f has distortion 3.*

Throughout the remainder of the paper, we assume that an election $\mathcal{E} = (V, C, \succ_V)$ is given, and we drop \mathcal{E} from notation when it is clear from the context.

3 Optimal Distortion via PLURALITYVETO

We now introduce PLURALITYVETO formally and show that it has distortion 3. PLURALITYVETO requires very limited ordinal information; it only requires one each of the following two types of queries to every voter.

- A *top query* to a voter v simply returns $\text{top}(v)$.
- A *bottom-among query* to a voter v regarding a subset of candidates A returns the bottom choice of v among candidates in A , denoted by $\text{bottom}_A(v)$.

Notice that both can of course be easily derived when each voter’s full ranking \succ_v is known.

PLURALITYVETO assigns an initial score of $\text{plu}(c)$ to each candidate c ; doing so only requires making a top query to each voter. These scores are then gradually decreased; when the score of a candidate c reaches zero, we say that c is *eliminated*. Voters are processed one by one in an arbitrary order; this order can be fixed beforehand, or be adaptive and

⁴We only care about the distances between voters and candidates, so d can be defined as a function $d : V \times C \rightarrow \mathbb{R}_{\geq 0}$ instead of on $V \cup C$. The triangle inequality can then be written as $0 \leq d(v, c) \leq d(v, c') + d(v', c') + d(v', c)$ for all $v, v' \in V$ and for all $c, c' \in C$.

based on voters' preferences. When a voter v is processed, she decrements the score of her bottom choice among the not-yet-eliminated candidates; the bottom choice can be found by making a bottom-among query to v . The winner is the last eliminated candidate. Pseudo-code is given as Algorithm 1.

Algorithm 1 PLURALITYVETO

Input: An election $\mathcal{E} = (V, C, \succ_V)$
Output: A winning candidate $c \in C$

- 1: **initialize** $\text{score}(c) = \text{plu}(c)$ for each $c \in C$
- 2: **let** (v_1, \dots, v_n) be an arbitrary ordering of V
- 3: **for** $i = 1, 2, \dots, n$ **do**
- 4: $A_i = \{c \in C : \text{score}(c) > 0\}$
- 5: $c_i = \text{bottom}_{A_i}(v_i)$
- 6: **decrement** $\text{score}(c_i)$ **by** 1
- 7: **return** c_n

Theorem 1. *The distortion of PLURALITYVETO is 3.*

Proof. We show that $G(c_n)$ has a perfect matching, which proves that PLURALITYVETO has distortion 3 by Lemma 1. Initially, the scores of candidates add up to n , and in each of the n iterations, a positive score is decremented by 1. Thus, the score of each candidate must be 0 at the end. This implies that, for each candidate c , there are $\text{plu}(c)$ distinct voters v_i such that $c_i = c$. In other words, for each voter v_i , we can define a unique voter v'_i such that $\text{top}(v'_i) = \text{bottom}_{A_i}(v_i)$. This means that v_i ranks any candidate in A_i weakly higher than $\text{top}(v'_i)$. Since $\text{score}(c_n)$ does not get to 0 until the end, note that $c_n \in A_i$ for all i . Hence, each voter v_i ranks c_n weakly higher than $\text{top}(v'_i)$, i.e., $(v_i, v'_i) \in E_{c_n}$. Thereby, we have shown that $G(c_n)$ has a perfect matching. \square

Algorithm 1 is not only quite natural; it also requires only $O(\log m)$ bits of information from each voter v_i , namely, $\text{top}(v_i)$ and $\text{bottom}_{A_i}(v_i)$. However, implementing the rule with these two queries comes with a trade-off: voters need to wait for possibly n rounds after reporting their top choice.

4 A Simple Proof of the Ranking-Matching Lemma via FRACTIONALVETO

Our proof of Theorem 1 implied that there always exists a candidate whose domination graph has a perfect matching. The Ranking-Matching Lemma in [Gkatzelis *et al.*, 2020] is a stronger existence result based on an extension of domination graphs in which nodes have arbitrary weights. Given an election $\mathcal{E} = (V, C, \succ_V)$ and weight vectors $\mathbf{p} \in \Delta(V)$ and $\mathbf{q} \in \Delta(C)$, the (\mathbf{p}, \mathbf{q}) -domination graph of a candidate $\tilde{c} \in C$ is the bipartite graph $G_{\mathbf{p}, \mathbf{q}}^{\mathcal{E}}(\tilde{c}) = (V, C, E_{\tilde{c}}, \mathbf{p}, \mathbf{q})$; the edge $(v, c) \in E_{\tilde{c}}$ if and only if $\tilde{c} \succ_v c$. A *fractional perfect matching* of $G_{\mathbf{p}, \mathbf{q}}^{\mathcal{E}}(\tilde{c})$ is a weight function $w : E_{\tilde{c}} \rightarrow \mathbb{R}_{\geq 0}$ such that $\sum_{c \in C: (v, c) \in E_{\tilde{c}}} w(v, c) = p_v$ for each voter v and $\sum_{v \in V: (v, c) \in E_{\tilde{c}}} w(v, c) = q_c$ for each candidate c . That is, the total weight of edges incident to each node must be equal to the weight of the node.

A candidate's domination graph has a perfect matching if and only if his $(\mathbf{p}^{\text{uni}}, \mathbf{q}^{\text{plu}})$ -domination graph has a fractional perfect matching; here, $p_v^{\text{uni}} = 1/n$ for all $v \in V$ and $q_c^{\text{plu}} = \text{plu}(c)/n$ for all $c \in C$. The Ranking-Matching Lemma gives a stronger existence guarantee by asserting that, for any election \mathcal{E} and any $\mathbf{p} \in \Delta(V)$ and $\mathbf{q} \in \Delta(C)$, there is a candidate $c \in C$ whose (\mathbf{p}, \mathbf{q}) -domination graph $G_{\mathbf{p}, \mathbf{q}}^{\mathcal{E}}(c)$ has a fractional perfect matching. Gkatzelis *et al.* [2020] give a proof by minimal counterexample for this lemma; thus, their proof is non-constructive. Moreover, it contains elaborate details for defining a minimal counterexample and intricate arguments involving smaller elections where some voters and candidates are removed.

We provide a much simpler and constructive proof of the Ranking-Matching Lemma. We generalize PLURALITYVETO in a way that allows any $\mathbf{p} \in \Delta(V)$ and $\mathbf{q} \in \Delta(C)$ to be given as input. We refer to this voting rule as FRACTIONALVETO, and give the pseudo-code as Algorithm 2. Notice that for the special case with weights \mathbf{p}^{uni} and \mathbf{q}^{plu} , FRACTIONALVETO specializes to PLURALITYVETO.

Algorithm 2 FRACTIONALVETO

Input: An election $\mathcal{E} = (V, C, \succ_V)$ along with normalized weight vectors $\mathbf{p} \in \Delta(V)$ and $\mathbf{q} \in \Delta(C)$
Output: A candidate $c \in C$ whose (\mathbf{p}, \mathbf{q}) -domination graph has a fractional perfect matching

- 1: **initialize** $\text{weight}(v) = p_v$ for each $v \in V$
- 2: **initialize** $\text{weight}(c) = q_c$ for each $c \in C$
- 3: **while** there is a $v \in V$ with $\text{weight}(v) > 0$ **do**
- 4: **let** v be such a voter
- 5: $A = \{c \in C : \text{weight}(c) > 0\}$
- 6: $c = \text{bottom}_A(v)$
- 7: $\epsilon = \min \{\text{weight}(v), \text{weight}(c)\}$
- 8: **decrement** $\text{weight}(v)$ **by** ϵ
- 9: **decrement** $\text{weight}(c)$ **by** ϵ
- 10: **return** c (i.e., the last candidate whose weight is decremented)

Theorem 2. *Given any election $\mathcal{E} = (V, C, \succ_V)$, and weight vectors $\mathbf{p} \in \Delta(V)$ and $\mathbf{q} \in \Delta(C)$, FRACTIONALVETO returns a candidate $c \in C$ whose (\mathbf{p}, \mathbf{q}) -domination graph $G_{\mathbf{p}, \mathbf{q}}^{\mathcal{E}}(c)$ has a fractional perfect matching.*

Proof. We begin by observing that the **while** loop (lines 3–9) terminates in at most $n + m$ iterations since in each iteration, the weight of either a voter v or a candidate c reaches 0. Next, we show that if a candidate, say \tilde{c} , wins, then $G_{\mathbf{p}, \mathbf{q}}(\tilde{c})$ has a fractional perfect matching $w : E_{\tilde{c}} \rightarrow \mathbb{R}_{\geq 0}$.

First, note that, since \tilde{c} is the last candidate whose weight is decremented, $\tilde{c} \in A$ for all iterations. Let us now consider an arbitrary iteration of FRACTIONALVETO in which a voter v is chosen. By definition, v ranks any candidate in A weakly higher than the candidate $c = \text{bottom}_A(v)$; in particular, v ranks \tilde{c} weakly higher than c , i.e., $(v, c) \in E_{\tilde{c}}$. Let w assign a weight of ϵ to the edge (v, c) , i.e., $w(v, c) = \epsilon$. We now show that w is a fractional perfect matching of $G_{\mathbf{p}, \mathbf{q}}(\tilde{c})$.

For each edge $(v, c) \in E_{\tilde{c}}$ to which w assigns positive weight, there exists an iteration where the weights of both

v and c are decremented by $w(v, c)$. Hence, the total weight of edges incident to each voter and candidate is equal to how much their weight is decremented until the end. All we need to show is that the weight of each voter v and candidate c reaches 0 at the end as they are initialized, respectively, to p_v and q_c . The total weights of voters and candidates are initially the same, and they stay so after each iteration since only a single voter's and candidate's weights are decremented, and by the same amount. Therefore, when the weight of each voter reaches 0, so does each candidate's. Thus, we have shown that w is a fractional perfect matching of $G_{p,q}(\tilde{C})$. \square

5 Randomized Voting Rules with Distortion 3

As shown in Theorem 1, PLURALITYVETO has distortion 3. Another voting rule that is well known to have distortion 3 (or $3 - 2/n$, to be precise) is RANDOMDICTATORSHIP, which returns the top choice of a voter chosen uniformly at random [Anshelevich and Postl, 2016]. In the same vein, one can view PLURALITYVETO as choosing a voter *deterministically* and returning that voter's top choice; this voter is referred to as v'_n in the proof of Theorem 1. This suggests a more general class of randomized voting rules, interpolating between RANDOMDICTATORSHIP and PLURALITYVETO: carefully rule out k voters, and return the top choice of a uniformly random voter among the remaining $n - k$ voters. In this section, we introduce such a general rule and show that it achieves distortion at most 3 for every choice of $k \in \{0, 1, \dots, n - 1\}$.

A randomized voting rule f is an algorithm which, given a ranked-choice profile \succ_V , returns a probability distribution over candidates $f(\succ_V) \in \Delta(C)$. Writing $\mathbf{w} = f(\succ_V)$, each candidate c is chosen as the winner by f with probability w_c . We refer to \mathbf{w} as the *winner distribution* of f . The cost of a winner distribution under a given metric is the expected cost of the winner, i.e., $\text{cost}(\mathbf{w}, d) = \sum_{c \in C} w_c \cdot \text{cost}(c, d)$. The distortion of a randomized voting rule is still the ratio of the rule's cost to the cost of the optimum solution. For randomized rules, the current best upper bound is $3 - 2/m$ [Kempe, 2020b], which is typically slightly better than the distortion of RANDOMDICTATORSHIP. This upper bound is known to be best possible among rules that randomize over only candidates with positive plurality score [Gkatzelis *et al.*, 2020]. It is open whether any randomized voting rule has distortion $3 - \Omega(1)$. The current best lower bound is 2.1126, due to Charikar and Ramakrishnan [2022].

A candidate c wins under RANDOMDICTATORSHIP with probability proportional to his plurality score $\text{plu}(c)$. When the score of candidate c_i is decremented at the i^{th} iteration of PLURALITYVETO, suppose that we rule out voter v'_i , as defined in the proof of Theorem 1. Since $\text{top}(v'_i) = c_i$, the current score of candidate c_i can be viewed as his plurality score with respect to the remaining voters v'_{i+1}, \dots, v'_n . Our generalized voting rule (Algorithm 3) therefore simply runs PLURALITYVETO for only $k < n$ iterations, then chooses a candidate with probability proportional to the residual scores.

Theorem 3. *The distortion of k -ROUNDPLURALITYVETO is at most 3 for any number of rounds $0 \leq k < n$.*

Our proof is based on a generalization of the flow technique from [Kempe, 2020a] to randomized voting rules. It is

Algorithm 3 k -ROUNDPLURALITYVETO

Input: An election $\mathcal{E} = (V, C, \succ_V)$
Output: A winner distribution $\mathbf{w} \in \Delta(C)$

- 1: **initialize** $\text{score}(c) = \text{plu}(c)$ for each $c \in C$
 - 2: **let** (v_1, \dots, v_n) be an arbitrary ordering of V
 - 3: **for** $i = 1, 2, \dots, k$ **do**
 - 4: $A_i = \{c \in C : \text{score}(c) > 0\}$
 - 5: $c_i = \text{bottom}_{A_i}(v_i)$
 - 6: **decrement** $\text{score}(c_i)$ **by** 1
 - 7: **return** $w_c = \text{score}(c)/(n - k)$ for all $c \in C$
-

encapsulated in Lemma 2 below; this lemma is a straightforward generalization of Lemma 3.1 from [Kempe, 2020a]. As in that paper, the proof is somewhat technical and long; thus, it is deferred to the full version.

The key concept, adopted from [Kempe, 2020a], is the following flow network. Given an election \mathcal{E} , let $H_{\mathcal{E}} = (V \times C, E)$ be a directed graph with the the following edges:

- For every voter v and any pair of candidates c and c' such that $c \succ_v c'$, there is a directed so-called *preference edge* $(v, c) \rightarrow (v, c')$ in E .
- For every candidate c and any pair $v \neq v'$ of distinct voters, there is a (bi-directed) so-called *sideways edge* $(v, c) \leftrightarrow (v', c)$ in E .

For a winner distribution⁵ $\mathbf{w} \in \Delta(C)$ and a candidate c^* , a (\mathbf{w}, c^*) -flow on $H_{\mathcal{E}}$ is a circulation g (i.e., non-negative and conserving flow unless specified otherwise) in which

- For each candidate c and voter v , exactly w_c units of flow originate at the node (v, c) .
- Flow is only absorbed at nodes (v, c^*) .

The *cost of g at voter v* is the total amount of flow absorbed at (v, c^*) , plus the total flow on sideways edges into or out of nodes (v, c) , for any candidate c . Formally, $\text{cost}_v(g) = \sum_{e \text{ into } (v, c^*)} g_e + \sum_{c \neq c^*} \sum_{v' \neq v} g_{(v', c) \leftrightarrow (v, c)}$ where $g_{u \leftrightarrow u'} = g_{u \rightarrow u'} + g_{u' \rightarrow u}$. The *cost of g* is $\text{cost}(g) = \max_{v \in V} \text{cost}_v(g)$. The key lemma showing how to use flows to upper-bound the distortion of a voting rule is the following.

Lemma 2. *Let f be a randomized voting rule. For every election $\mathcal{E} = (V, C, \succ_V)$ and any candidate $c^* \in C$, suppose that there is a (\mathbf{w}, c^*) -flow g on the flow network $H_{\mathcal{E}}$ such that $\mathbf{w} = f(\succ_V)$ and $\text{cost}(g) \leq \lambda$. Then, $\text{dist}(f) \leq \lambda$.*

Proof of Theorem 3. Fix an arbitrary candidate $c^* \in C$. We will describe a (\mathbf{w}, c^*) -flow g , and prove that it has cost at most 3. Theorem 3 then follows directly from Lemma 2.

As shown in the proof of Theorem 1, we can sort the voters as v'_1, \dots, v'_n such that $\text{top}(v'_i) = \text{bottom}_{A_i}(v_i)$ for all i in the execution of PLURALITYVETO, i.e., without stopping early. Define $V_k = \{v_1, \dots, v_k\}$ and $\bar{V}_k = \{v_{k+1}, \dots, v_n\}$; also $V'_k = \{v'_1, \dots, v'_k\}$ and $\bar{V}'_k = \{v'_{k+1}, \dots, v'_n\}$.

⁵Kempe [2020a] considered only deterministic voting rules; in that case, the distribution \mathbf{w} was restricted to have probability 1 for the deterministic winner, and 0 for all other candidates.

In order to obtain a (w, c^*) -flow, we must route flow such that for each candidate c and each voter $v \in V$, exactly $\text{score}(c)/(n-k)$ units of flow originate at the node (v, c) . In particular, because $\text{score}(c) = 0$ for candidates $c \notin A_k$, no flow originates at any nodes (v, c) for $c \notin A_k$.

1. First, consider a voter $v_i \in V_k$, so that $i \leq k$. Since $A_k \subseteq A_i$, voter v_i ranks all candidates in A_k weakly higher than $\text{bottom}_{A_i}(v_i)$. Therefore, g can route all the flow originating at nodes (v_i, c) to $(v_i, \text{bottom}_{A_i}(v_i))$ along preference edges. At that point, there is one unit of flow at $(v_i, \text{bottom}_{A_i}(v_i))$. This one unit of flow is next routed to $(v'_i, \text{bottom}_{A_i}(v_i)) = (v'_i, \text{top}(v'_i))$ using a sideways edge. Finally, since $\text{top}(v'_i)$ is by definition the top choice of v'_i , the unit can be routed to (v'_i, c^*) using a preference edge, and is then absorbed.
2. Next, we (jointly) consider all voters $v_i \in \bar{V}_k$. Fix a candidate $c \in A_k$. Because $\text{score}(c)/(n-k)$ units of flow originate at each node (v_i, c) for $v_i \in \bar{V}_k$, and there are $|\bar{V}_k| = n-k$ such nodes, in total, exactly $\text{score}(c)$ units of flow originate at these nodes. On the other hand, $\text{score}(c)$ is also the number of distinct voters $v'_j \in \bar{V}'_k$ such that $\text{top}(v'_j) = c$. g distributes all the $\text{score}(c)$ units of flow from nodes (v_i, c) (for $v_i \in \bar{V}_k$) to the $\text{score}(c)$ nodes (v'_j, c) with $\text{top}(v'_j) = c$, in a way that each voter $v'_j \in \bar{V}'_k$ receives one unit of flow. Then, for each voter $v'_j \in \bar{V}'_k$ with $\text{top}(v'_j) = c$, the flow g routes the one unit of flow from (v'_j, c) to (v'_j, c^*) using a preference edge; there, it is absorbed. Because flow only originates at nodes (v_i, c) with $c \in A_k$, all the flow is in fact routed to an absorbing node in this way.

In the flow g described above, for each voter v , exactly one unit is sent out on sideways edges, and one unit is received on sideways edges. For voters in V_k , the unit sent is by the first case above; for voters in V'_k , the unit received is by the first case. For voters in \bar{V}_k , the unit sent is by the second case; for voters in \bar{V}'_k , the unit received is by the second case. Finally, each voter v_i absorbs the one received unit of flow at (v_i, c^*) . This proves that $\text{cost}(g) = 3$, completing the proof. \square

6 Conclusion and Future Work

We showed that a simple deterministic voting rule, called PLURALITYVETO, achieves the optimal metric distortion of 3; the proof is short and simple. We used a generalization to prove the Ranking-Matching Lemma of Gkatzelis *et al.* [2020], and we also showed that a class of randomized rules interpolating between RANDOMDICTATORSHIP and PLURALITYVETO all achieve expected distortion at most 3.

PLURALITYVETO is a very natural and potentially practical voting rule, and it would be of interest to understand which other properties it satisfies. One difficulty is that the outcome depends on the order in which the voters are queried in the multi-round veto process. On the other hand, this makes the structure of the set $W \subseteq C$ of potential winners (i.e., candidates who will win for at least one processing order of the voters) an interesting object of study.

Note that W is a subset of candidates whose domination graph has a perfect matching, i.e., potential winners of PLURALITYMATCHING. This has several immediate implications. First, it is possible that for all orderings of voters, the selected candidate fails to be a Condorcet winner, as shown for PLURALITYMATCHING by Gkatzelis *et al.* [2020]. Second, if the metric space is α -decisive (i.e., for each voter, the ratio between her distance to her top-ranked and second-ranked candidate is at most α [Anshelevich and Postl, 2016]) then the distortion bound improves to $2 + \alpha$.

Additional insights can be gleaned from the *Peer Selection* setting, in which the set of voters is the same as the set of candidates, so each voter/candidate ranks herself first; as a result, all such instances are 0-decisive. PLURALITYVETO can be even more easily described in the Peer Selection setting: in each round i , the voter v_i eliminates from consideration the remaining candidate furthest from her.

Here, we consider a slight variant of PLURALITYVETO: the order of voters is chosen adaptively, and the next voter v_{i+1} is always one whose first-place vote was canceled by v_i . In addition, the vote of v_1 is always canceled, so the winner is the top choice of v_n . A proof essentially identical⁶ to the one of Theorem 1 shows that this rule returns a candidate whose domination graph has a perfect matching as well. In turn, we can use this insight to prove that there are at least *two* candidates who can win in Peer Selection, i.e., $|W| \geq 2$. First, if the process is run from an arbitrary v_1 , then some $\hat{v} = v_n$ wins. If the process is run starting from $v_1 = \hat{v}$, the winner must be some other candidate $\tilde{v} \neq \hat{v}$, because \hat{v} is eliminated in the first step. So there are at least two potential winners.

Peer selection also has interesting properties when the voters/candidates are embedded in Euclidean space \mathbb{R}^D . Since each voter, on her turn, eliminates the voter furthest from her, the eliminated voter is always located at a corner of the convex hull of all previously uneliminated ones. In this sense, PLURALITYVETO “peels away” extreme candidates one by one⁷: the convex hull of the voters in A_k (who are not yet eliminated after k rounds) contains no voter from \bar{A}_k . A natural question is whether the set W of potential winners has the same convexity property, i.e., the convex hull of W contains no voter/candidate from $C \setminus W$.⁸ If true, this would show that the potential winners are in a sense “cohesive.”

Another compelling direction concerns the incentives under PLURALITYVETO. We described it as a sequential process in which voters are queried about their bottom choice one by one. In describing the sequential process, we assumed that all queries are answered truthfully. An interesting direction here is to consider the “Price of Anarchy:” what is the worst distortion of PLURALITYVETO if agents play a subgame perfect equilibrium in the VETO stage instead of truthfully revealing their bottom choice?

⁶Because the vote of v_1 is always canceled, this rule is not technically a special case of PLURALITYVETO.

⁷This behavior also provides some informal intuition for why the final candidate should be close to the geometric median.

⁸For the general case (rather than Peer Selection), this is false: for example, if there is one voter and candidate each on the left and right, but only one candidate (with no voters) in the center, then the left and right candidates can win, but not the center one.

References

- [Anshelevich and Postl, 2016] Elliot Anshelevich and John Postl. Randomized social choice functions under metric preferences. In *Proc. 25th Intl. Joint Conf. on Artificial Intelligence*, pages 46–59, 2016.
- [Anshelevich *et al.*, 2015] Elliot Anshelevich, Onkar Bhardwaj, and John Postl. Approximating optimal social choice under metric preferences. In *Proc. 29th AAAI Conf. on Artificial Intelligence*, pages 777–783, 2015.
- [Anshelevich *et al.*, 2018] Elliot Anshelevich, Onkar Bhardwaj, Edith Elkind, John Postl, and Piotr Skowron. Approximating optimal social choice under metric preferences. *Artificial Intelligence*, 264:27–51, 2018.
- [Anshelevich *et al.*, 2021a] Elliot Anshelevich, Aris Filos-Ratsikas, Nisarg Shah, and Alexandros A. Voudouris. Distortion in social choice problems: an annotated reading list. *SIGecom Exchanges*, 19(1):12–14, 2021.
- [Anshelevich *et al.*, 2021b] Elliot Anshelevich, Aris Filos-Ratsikas, Nisarg Shah, and Alexandros A. Voudouris. Distortion in social choice problems: The first 15 years and beyond. In *Proc. 30th Intl. Joint Conf. on Artificial Intelligence*, pages 4294–4301, 2021.
- [Black, 1948] Duncan Black. On the rationale of group decision making. *J. Political Economy*, 56:23–34, 1948.
- [Boutilier and Rosenschein, 2016] Craig Boutilier and Jeffrey S. Rosenschein. Incomplete information and communication in voting. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 10, pages 223–257. Cambridge University Press, 2016.
- [Boutilier *et al.*, 2015] Craig Boutilier, Ioannis Caragiannis, Simi Haber, Tyler Lu, Ariel D. Procaccia, and Or Sheffet. Optimal social choice functions: A utilitarian view. *Artificial Intelligence*, 227:190–213, 2015.
- [Caragiannis and Procaccia, 2011] Ioannis Caragiannis and Ariel D. Procaccia. Voting almost maximizes social welfare despite limited communication. *Artificial Intelligence*, 175(9):1655–1671, 2011.
- [Caragiannis *et al.*, 2017] Ioannis Caragiannis, Swaprava Nath, Ariel D Procaccia, and Nisarg Shah. Subset selection via implicit utilitarian voting. *J. Artif. Intell. Res.*, 58:123–152, 2017.
- [Charikar and Ramakrishnan, 2022] Moses Charikar and Prasanna Ramakrishnan. Metric distortion bounds for randomized social choice. In *Proc. 33rd ACM-SIAM Symp. on Discrete Algorithms*, 2022. to appear. Preprint available at <https://arxiv.org/abs/2111.03694>.
- [Fain *et al.*, 2019] Brandon Fain, Ashish Goel, Kamesh Munagala, and Nina Prabhu. Random dictators with a random referee: Constant sample complexity mechanisms for social choice. In *Proc. 33rd AAAI Conf. on Artificial Intelligence*, pages 1893–1900, 2019.
- [Gkatzelis *et al.*, 2020] Vasilis Gkatzelis, Daniel Halpern, and Nisarg Shah. Resolving the optimal metric distortion conjecture. In *Proc. 61st IEEE Symp. on Foundations of Computer Science*, pages 1427–1438, 2020.
- [Goel *et al.*, 2017] Ashish Goel, Anilesh Kollagunta Krishnaswamy, and Kamesh Munagala. Metric distortion of social choice rules: Lower bounds and fairness properties. In *Proc. 18th ACM Conf. on Economics and Computation*, pages 287–304, 2017.
- [Kempe, 2020a] David Kempe. An analysis framework for metric voting based on LP duality. In *Proc. 34th AAAI Conf. on Artificial Intelligence*, pages 2079–2086, 2020.
- [Kempe, 2020b] David Kempe. Communication, distortion, and randomness in metric voting. In *Proc. 34th AAAI Conf. on Artificial Intelligence*, pages 2087–2094, 2020.
- [Moulin, 1980] Hervé Moulin. On strategy-proofness and single peakedness. *Public Choice*, 35:437–455, 1980.
- [Munagala and Wang, 2019] Kamesh Munagala and Kangning Wang. Improved metric distortion for deterministic social choice rules. In *Proc. 20th ACM Conf. on Economics and Computation*, pages 245–262, 2019.
- [Procaccia and Rosenschein, 2006] Ariel D. Procaccia and Jeffrey S. Rosenschein. The distortion of cardinal preferences in voting. In *Proc. 10th Intl. Workshop on Cooperative Inform. Agents X*, pages 317–331, 2006.
- [Procaccia, 2010] Ariel D. Procaccia. Can approximation circumvent Gibbard-Satterthwaite? In *Proc. 24th AAAI Conf. on Artificial Intelligence*, pages 836–841, 2010.
- [Seeley, 2010] Thomas D. Seeley. *Honeybee Democracy*. Princeton University Press, Princeton, N.J., 2010.