Outer Diversity of Structured Domains

Piotr Faliszewski
AGH University
Kraków, Poland
faliszew@agh.edu.pl

Krzysztof Sornat
AGH University
Kraków, Poland
sornat@agh.edu.pl

Stanisław Szufa
University of Geneva
Geneva, Switzerland
s.szufa@gmail.com

Tomasz Wąs
University of Oxford
Oxford, United Kingdom
tomasz.was@cs.ox.ac.uk

ABSTRACT

An ordinal preference domain is a subset of preference orders that the voters are allowed to cast in an election. We introduce and study the notion of *outer diversity* of a domain and evaluate its value for a number of well-known structured domains, such as the single-peaked, single-crossing, group-separable, and Euclidean ones.

KEYWORDS

Diversity, Ordinal Elections, Structured Domains, Single-Peaked, Single-Crossing.

ACM Reference Format:
Piotr Faliszewski, Krzysztof Sornat, Stanisław Szufa, and Tomasz Wąs. 2026. Outer Diversity of Structured Domains. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/WNGL6996

1 INTRODUCTION

In the standard, ordinal model of elections, each voter considers a set of candidates and ranks them from the one that he or she likes most to the one that he or she likes least. In principle, a voter may order the candidates in any arbitrary way, but some of these rankings appear more natural (or, more rational) than others. For example, in the political setting it would be expected that a voter would rank the candidates with respect to their proximity to his or her political stance, but a ranking with the most right-wing candidate and the most left-wing one on two top positions would be surprising. Various rationality conditions for ordinal rankings are expressed as so-called *structured domains*, i.e., sets of rankings that can be cast in a given setting. Such domains include, e.g., the single-peaked one [4], which captures preferences based on proximity to some ideal, the single-crossing ones, introduced in the context of taxation [28, 31], or group-separable ones [22, 23], where voters derive rankings of candidates from preferences over their features [15, 24]. We introduce a new measure of diversity of such domains, provide algorithms for computing its value, and analyze diversity of a number of structured domains.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. https://doi.org/10.65109/WNGL6996

Somewhat surprisingly, analysis of diversity for structured domains has only recently started to receive more focused attention [1, 18, 25], with a few authors also considering diversity of elections [14, 16, 20]. Two commonly used approaches are:

Richness Diversity. The overarching idea is that a domain is diverse if it contains many different substructures in its rankings (these substructures are sometimes also called *attributes*, as the approach builds on the theory of attribute diversity of Nehring and Puppe [29]). For example, one might consider how many votes appear in the domain, how many candidates are ever ranked on top, or—for each triple of candidates—how many ways of ranking these candidates appear in the domain. This approach is taken, e.g., by Ammann and Puppe [1] and Karpov et al. [25]

Inner Diversity. In this case, we say that a domain is diverse if its rankings do not form clear clusters. This approach was taken by Faliszewski et al. [14, 16, 18], who introduced the k-Kemeny problem to quantify the difficulty of clustering rankings (briefly put, one tries to optimally partition the rankings into a given number of groups, measuring their cohesiveness using the classic Kemeny rule [26]).

We propose a third approach, which we refer to as *outer diversity*:

Outer Diversity. A domain is diverse if, on average, a random ranking from the space of all possible ones is similar to some ranking from the domain. In particular, we measure similarity between rankings using the number of swaps of adjacent candidates that transform one into the other.

Inner and outer diversity seem to capture the same basic intuition, but the inner approach focuses on the rankings within the domain, whereas the outer one focuses on those outside.

We believe that all the above approaches to measuring domain diversity are meaningful and are worth studying, but outer diversity has some advantages. First, it has a very clear interpretation: If a domain has high diversity, then it covers the space of all possible rankings well; if one wanted to cast a ranking from the domain but had one that did not belong to it, then the closest member of the domain would not be too far off from his or her original ranking.

Second, outer diversity of a given domain is a single number. On the other hand, in case of richness diversity one has to choose from many different substructures to count, and in case of inner diversity one either has to choose the number of clusters to consider (for which there is no clear solution) or somehow aggregate obtained values for different numbers of clusters, which is not obvious (indeed, the works we cite with respect to inner diversity do not provide fully satisfying recommendations).

Third, while in principle computing outer diversity may require exponential time, we provide efficient algorithms for computing it using sampling: Our algorithms compute the distance from a given ranking to the closest one in a given domain of interest, such as the single-peaked, single-crossing, and group-separable ones. Hence, we can sample random votes, compute their distances to the domains, and output the average of the obtained values. On the other hand, even the heuristics that Faliszewski et al. [14] proposed for inner diversity (i.e., for k-Kemeny) require exponential time if a given domain contains exponentially many rankings (as is the case for, e.g., the single-peaked and group-separable ones).

Our main contributions are as follows:

(1) We introduce the notion of outer diversity and provide means of computing its values for a number of domains, including the single-peaked, single-crossing, and group-separable ones, but also many others (including variants of the single-peaked domain, as well as Euclidean domains). However, we also find that for some natural domains, the sampling-based approach requires solving an NP-hard problem.
(2) We evaluate outer diversity across a number of domains. We find that ranking the domains with respect to outer diversity gives similar results as doing so with respect to the inner one. Further, while analyzing outer diversity of our domains, we note a number of their interesting features.
(3) We compute domains of given sizes, whose outer diversity is (close to) the highest possible, and we analyze how close are various structured domains to these maximal values.

One of the takeaway messages of our work is that the domain of group-separable preferences based on caterpillar trees (see Section 2) is the most diverse one among those that we study, and has many features that other domains often lack. Consequently, and strengthening the message of Faliszewski et al. [18], we believe that this domain should be used in numerical experiments on elections. Even if it does not capture reality in a given setting, it is so special that studying it may lead to the discovery of hard-to-spot phenomena.

Omitted proofs are available in the full version of the paper [19].

2 PRELIMINARIES

For a positive integer t, by $[t]$ we mean the set $\{1, 2, \dots, t\}$. Given an undirected graph G, by $V(G)$ and $E(G)$ we mean its sets of vertices and edges, respectively. We use the *Iverson bracket* notation, i.e., for a logical formula φ, by $[\varphi]$ we mean 1 if φ is true, and 0, otherwise.

2.1 Preference Orders, Domains, and Elections

Let C be a set of m *candidates*. By $\mathcal{L}(C)$ we denote the set of all $m!$ linear orders over C, typically referred to as *preference orders*, *votes*, or *rankings*. For each such ranking v and two candidates $a, b \in C$, we write $a \succ_v b$ to indicate that v ranks a ahead of b (i.e., according to v, a is preferred to b). A *preference domain* (over C) is a subset D of $\mathcal{L}(C)$. In particular, $\mathcal{L}(C)$ is the *general domain*. For a ranking v and candidate c, by $\mathrm{pos}_v(c)$ we mean the position of c in v; the top candidate has position 1, the next one has position 2, and so on.

An election is a pair $E = (C, V)$, where $C = \{c_1, \dots, c_m\}$ is a set of candidates and $V = (v_1, \dots, v_n)$ is a collection of voters, each of whom has a vote from $\mathcal{L}(C)$. To streamline the discussion, we use the same symbol v_i to refer both to the given voter and to his or her vote. The exact meaning will always be clear from the context. Given a domain $D \subseteq \mathcal{L}(C)$, we say that $E = (C, V)$ is a D-election if all the voters in V have votes from D.

Sometimes it is convenient to treat a domain $D \subseteq \mathcal{L}(C)$ as an election that contains a single voter for each of its preference orders. In particular, we write UN to mean an election that contains one copy of every possible order (so UN is simply $\mathcal{L}(C)$, viewed as an election). For other domains, we typically do not introduce a second name, but UN has already been used in preceding literature in the context of the map of elections [32].

For two rankings $u, v \in \mathcal{L}(C)$, their *swap distance*[1] is a number of pairs of candidates in C on whose ordering u and v disagree, i.e.:

$$\mathrm{swap}(u, v) = |\{a, b \in C : a \succ_u b \wedge b \succ_v a\}|.$$

The value $\mathrm{swap}(u, v)$ can be computed in time $O(m\sqrt{\log m})$ [6]. For a domain $D \subseteq \mathcal{L}(C)$, we let $\mathrm{swap}(D, v) = \min_{u \in D} \mathrm{swap}(u, v)$.

2.2 Structured Domains

Let us fix a size-m set of candidates $C = \{c_1, \dots, c_m\}$. Below, we describe the preference domains over C whose diversity we want to analyze.

Consider a connected, undirected graph G, such that $V(G) = C$ (we refer to such graphs as SP-graphs, or SP-trees in case G is also acyclic). A ranking $v \in \mathcal{L}(C)$ is *single-peaked* with respect to G if for every $t \in [m]$, the subgraph induced by the t top-ranked candidates from v is connected. $\mathrm{SP}(G)$ is the domain that consists of all rankings that are single-peaked with respect to G (see, e.g., the work of Elkind et al. [10]). We focus on the following variants:

(1) SP is the classic single-peaked domain that consists of rankings single-peaked with respect to a path (often called an *axis* and denoted $c_1 \rhd c_2 \rhd \cdots \rhd c_m$). In politics, the axis may, e.g., indicate the progression from the most left-wing candidate to the most right-wing one. SP is due to Black [4].
(2) SPOC, introduced by Peters and Lackner [30], consists of rankings single-peaked with respect to a cycle. SPOC preferences appear, e.g., when people located in different time zones want to choose a convenient time for an online meeting. The name SPOC stands for *single-peaked on a circle.*
(3) SP/DF is a domain introduced by Faliszewski et al. [18] and consists of votes single-peaked with respect to a tree that we obtain by taking a path and adding four vertices: two directly connected to one end of the path, and two directly connected to the other end. The name SP/DF stands for *single-peaked/double-forked.* Domains of rankings single-peaked with respect to trees were introduced by Demange [7].

Whenever we speak of SP, SPOC, or SP/DF the exact number of candidates and their positions in respective graphs will be clear from the context (or will be irrelevant). We use this convention for the other domains as well, omitting such details from their names.

[1] We focus on swap distance in this paper, as it is standard in (computational) social choice and it was also used in the work on inner diversity [14]. Our initial experiments with Spearman footrule distance [9] yield qualitatively similar outcomes.

A domain is *single-crossing* if it is possible to list its members as $v_1, v_2, \dots, v_n$, so that, as we consider them from v_1 to v_n, the relative ordering of each pair of candidates a and b changes at most once. Single-crossingness is due to Mirrlees [28] and Roberts [31].

(4) By SC, we mean a single-crossing domain sampled from the space of all such domains using the algorithm of Szufa et al. [32]: We generate votes iteratively, starting with some arbitrary vote v_0. In each iteration, given vote v_i, we form v_{i+1} by taking v_i's copy and swapping a randomly selected pair of adjacent candidates that were not swapped in preceding iterations. Altogether, we generate rankings $v_0, \dots, v_{\binom{m}{2}}$ that form our domain.

Note that the algorithm of Szufa et al. [32] does not sample single-crossing domains uniformly at random (so far, the only known algorithm for such uniform sampling requires exponential time).

Let T be an ordered, rooted tree, where each internal node has at least two children and each leaf is labeled with a unique candidate from C (we refer to such trees as GS-trees). A *frontier* of T is the ranking of the candidates, obtained by reading the leaves of T from left to right. Domain $\mathrm{GS}(T)$ consists exactly of those rankings $v \in \mathcal{L}(C)$ that are either a frontier of T or a frontier of a tree obtained from T by reversing the order of some nodes' children. A domain D is *group-separable* if $D = \mathrm{GS}(T)$ for some T. We are particularly interested in the following two such domains:

(5) GS/bal is a group-separable domain defined by balanced binary trees, i.e., binary trees where each internal node has exactly two children and for each two leaves, their distance from the root differs at most by 1.
(6) GS/cat is a group-separable domain defined by caterpillar binary trees, i.e., trees where each internal node has exactly two children, of which at least one is a leaf.

Group-separable domains were introduced by Inada [22, 23], but the above tree-based definition is due to Karpov [24].

Let d be some positive integer, and let $x\colon C \to \mathbb{R}^d$ be a function that associates the candidates with distinct points in $\mathbb{R}^d$. A ranking $v \in \mathcal{L}(C)$ is consistent with x if there is a point $x_v \in \mathbb{R}^d$ such that for each two candidates $a, b \in C$ such that $a \succ_v b$ it holds that the Euclidean distance between x_v and $x(a)$ is smaller than that between x_v and $x(b)$. $D(x)$ is the domain that includes exactly the rankings consistent with x. Such domains are called *Euclidean* and were studied, e.g., by Enelow and Hinich [11, 12]. We focus on:

(7) 1D-Int., 2D-Square, and 3D-Cube, where the position of each candidate is sampled uniformly at random from, respectively, $[-1, 1]$, $[-1, 1]^2$, and $[-1, 1]^3$.

It is well-known that 1D-Int. is also a single-crossing domain, and all its votes are single-peaked with respect to the axis obtained by sorting the positions of the candidates.

SP, SC, all group-separable domains, and 1D-Int. are examples of so-called *Condorcet domains*. That is, for every election with odd number of votes from one of these domains, there is a ranking v of the candidates such that if v ranks some candidate a over some other candidate b, then a strict majority of voters prefers a to b.

2.3 Distance Between Elections

Isomorphic swap distance between two elections (with the same numbers of candidates and the same numbers of voters) is a measure of their structural similarity, introduced by Faliszewski et al. [17]. We extend it to apply to elections with different numbers of voters (in essence, we pretend to duplicate the votes so that the elections appear to be equal-sized).

Definition 2.1. For two elections $E = (C, V)$ and $F = (B, U)$ such that $|C| = |B|$, where $V = (v_1, \dots, v_n)$ and $U = (u_1, \dots, u_k)$, their isomorphic swap distance is defined as follows (the indices of the votes from V are taken modulo n, and the indices of the votes from U are taken modulo k):

$$d_{\mathrm{swap}}(E, F) = \frac{1}{nk} \min_{\pi\colon [nk] \to [nk]} \min_{\sigma\colon C \to B} \sum_{i \in [nk]} \mathrm{swap}(\sigma(v_i), u_{\pi(i)}),$$

where π and σ are bijections, and by $\sigma(v_i)$ we mean vote v_i where each candidate $c \in C$ is replaced with candidate $\sigma(c) \in B$.

2.4 k-Kemeny and Inner Diversity

Let $E = (C, V)$ be an election and let $R = \{r_1, \dots, r_k\}$ be a set of preference orders from $\mathcal{L}(C)$. By the Kemeny score of R with respect to election E, we mean:

$$\mathrm{kem}_E(R) = \textstyle\sum_{v \in V} \mathrm{swap}(R, v).$$

In other words, it is the sum of the swap distances of the election's votes to their closest rankings from R. The k-Kemeny score of an election E, denoted k-kem(E), is the smallest Kemeny score of a size-up-to-k set of rankings for this election. By Kemeny score we mean the 1-Kemeny score. Computing the Kemeny score of a given election is well-known to be hard [2, 21], even for the case of four voters [3, 8]. The notion of the Kemeny score was the original idea of Kemeny [26], whereas the extension to collections of rankings was put forward by Faliszewski et al. [14], in the context of election diversity. Specifically, they claimed that the appropriately normalized weighted sum of an election's k-Kemeny scores (for varying k) captures its diversity. Indeed, the larger an election's k-Kemeny score, the more difficult it is to cluster its votes into k groups, meaning that its votes are quite different from one another. Consequently, these votes are diverse. The same view was taken by Faliszewski et al. [16] and was recently applied to measure the diversity of preference domains by Faliszewski et al. [18]. Specifically, given domain D over size-m candidate set, they defined its Kemeny vector to be:

$$\mathrm{kem}(D) = (1\text{-kem}(D)/|D|, 2\text{-kem}(D)/|D|, \dots, m\text{-kem}(D)/|D|)$$

and they said that a given domain D_1 is more diverse than another domain D_2 (both over equal-sized candidate sets) if $\mathrm{kem}(D_1)$ dominates $\mathrm{kem}(D_2)$ or is close to dominating it; they did not formalize this notion as they considered only a few domains.

We broadly refer to measures of diversity based on the difficulty of clustering as capturing *inner diversity*.

3 OUTER DIVERSITY

Let C be a set of candidates and let $D \subseteq \mathcal{L}(C)$ be a domain over C. By the *average normalized swap distance* of D, denoted $\mathrm{ansd}(D)$, we mean the expected swap distance between a vote chosen from

Table 1: For each domain we give its size and the complexity of finding its closest member (in terms of swap distance) to a given input ranking. Running times marked with * do not include the time needed for preprocessing.

Domain	Size	Complexity of Finding Closest Ranking from D
GS(T)	$\leq 2^{m-1}$	$O(m^2)$
GS/cat	2^{m-1}	$O(m \log m)$
GS/bal	2^{m-1}	$O(m \log m)$
SP	2^{m-1}	$O(m^2)$
SP/DF	$2^{m+1} - 16$	$O(m^4)$
SPOC	$m2^{m-2}$	$O(m^2)$
SP(T)	—	$O(km^k)$ k = number of T's leaves
SP(G)	—	NP-com.
SC	$1 + m(m-1)/2$	$O(m^2)^*$
1D-Int.	$1 + m(m-1)/2$	$O(m^2)^*$
2D-Square	$O(m^4)$	$O(m^4)^*$
3D-Cube	$O(m^6)$	$O(m^6)^*$

$\mathcal{L}(C)$ uniformly at random and the closest vote in D, divided by the maximal possible distance between two votes in $\mathcal{L}(C)$. Formally:

$$\mathrm{ansd}(D) = \tfrac{1}{m!} \textstyle\sum_{u \in \mathcal{L}(C)} \mathrm{swap}(D, u)/\binom{m}{2}.$$

The largest possible value of ansd(D) is 0.5, obtained when D consists of a single vote, and the smallest one is 0, obtained for the general domain. To ensure that *outer diversity* of a domain D is between 0 and 1 (where 0 means complete lack of diversity and 1 means full diversity), we define it as the following linear transformation of ansd(D).

Definition 3.1. For a domain $D \subseteq \mathcal{L}(C)$, its *outer diversity* is defined as out-div(D) = $1 - 2 \cdot \mathrm{ansd}(D)$.

While outer- and inner diversity notions are based on different principles, they are interrelated in several ways. For example, inner diversity, as defined by Faliszewski et al. [14, 16, 18], relies on analyzing k-Kemeny scores of given elections or domains, whereas ansd(D) is simply the normalized k-Kemeny score of the input domain D, with respect to the UN election. Considered from a different perspective, ansd(D) is equal to the smallest possible isomorphic swap distance between UN and a D-election.

Proposition 3.2. *For every domain $D \subseteq \mathcal{L}(C)$, it holds that* $\mathrm{ansd}(D) = \min_{E \text{ is a } D\text{-election}} d_{\mathrm{swap}}(\mathrm{UN}, E)/\binom{m}{2}$.

Since Faliszewski et al. [14] have shown that proximity to UN is highly correlated with their form of inner diversity, we conclude that both approaches are capturing the same high-level idea.

4 COMPUTING OUTER DIVERSITY

For domains over sufficiently small candidate sets, it is possible to compute outer diversity exactly. In the most basic approach, given a domain D over candidate set C, we could simply compute the swap distance between every vote in D and every vote in $\mathcal{L}(C)$. Naturally, this is very inefficient and computing outer diversity of, say, SP with m candidates would require time $O(m! \cdot 2^{m-1} \cdot m\sqrt{\log m})$; the general domain has $m!$ rankings, SP has 2^{m-1} of them, and it takes $O(m\sqrt{\log m})$ time to compute the swap distance [6]. Fortunately, there is a faster approach that given a domain D, for each i forms a set D_i of rankings at swap distance i from D.

Proposition 4.1. *There is an algorithm that given domain D over m candidates (represented by listing its members), computes* out-div(D) *in time* $O(m^2 \cdot m!)$.

To compute outer diversity for larger candidate sets, we resort to sampling. Namely, given a domain D over a size-m candidate set C, we fix a number N, sample N rankings from $\mathcal{L}(C)$, for each sampled ranking v we compute swap(D, v) and output the average of these values, divided by $\binom{m}{2}$. This gives an estimate for ansd(D), based on which we obtain out-div(D). However, to implement this idea efficiently, we need fast algorithms for the following problem: Given a ranking v and a domain D, compute swap(D, v). We dedicate the rest of this section to seeking algorithms for this problem for various domains, and to establishing its complexity.

On the outset, the problem can be even NP-hard. For example, for each set of $4m$ candidates $C = \{c_{i,j} : i \in [4], j \in [m]\}$, let the *4-alignment* domain contain each vote of the form $\{c_{1,1}, \dots, c_{1,m}\} \succ \{c_{2,1}, \dots, c_{2,m}\} \succ \{c_{3,1}, \dots, c_{3,m}\} \succ \{c_{4,1}, \dots, c_{4,m}\}$, in which the order of the candidates, based on their second indices, is identical in each block. Then we have the following hardness result (in essence, for this domain the problem of finding a closest vote in the domain becomes the problem of computing Kemeny score for 4 voters, known to be NP-hard [3, 8]).

Theorem 4.2. *Let D be the 4-alignment domain. Given vote v and integer $d \in \mathbb{N}$ it is NP-complete to decide whether* swap(D, v) $\leq d$.

Despite this negative result, for most of our domains we find efficient algorithms for computing the distance to a given vote (see Table 1). In the following, we always use $C = \{c_1, \dots, c_m\}$ to denote the set of m candidates in the domain under consideration.

4.1 Single-Peaked Domains

Let us first consider the family of single-peaked domains. We note that Faliszewski et al. [13, Theorem 4.5.] already gave a polynomial-time algorithm for computing the distance between SP and a given ranking, but their approach—based on dynamic programming—required $O(m^3)$ time. We improve this algorithm to run in $O(m^2)$ time. The main idea is to use dynamic programming to iteratively compute the distance between a given ranking v and votes that rank more and more bottom candidates as required by SP.

Assume that we are given a vote v and a societal axis $c_1 \rhd c_2 \rhd \cdots \rhd c_m$. For each $\ell, r \in \{0, 1, 2, \dots, m\}$ such that $\ell + r \leq m$, let $C_{\ell,r}$ denote the set of the first ℓ and the last r candidates according to $\rhd$. Formally, we have $C_{\ell,r} = \{c_1, \dots, c_\ell\} \cup \{c_{m+1-r}, \dots, c_m\}$; by convention, for $\ell = 0$ we have $\{c_1, \dots, c_\ell\} = \emptyset$, and for $r = 0$ we have $\{c_{m+1-r}, \dots, c_m\} = \emptyset$. Then, by $U_{\ell,r}$ we denote the set of all votes $u \in \mathcal{L}(C)$ in which (a) candidates from $C_{\ell,r}$ are in the bottom $\ell + r$ positions, and (b) for each $t \in \{m, m-1, \dots, m-\ell-r+1\}$, the top t candidates of u form an interval within $\rhd$. Observe that $U_{0,0} = \mathcal{L}(C)$, whereas if $\ell + r = m$, then $U_{\ell,r}$ = SP. We write $A_{\ell,r}$ to denote the minimal swap distance between v and $u \in U_{\ell,r}$.

Algorithm 1 Distance between a ranking and SP

Input: Ranking $v \in \mathcal{L}(C)$, societal axis $c_1 \rhd \cdots \rhd c_m$
  Phase 1, Precomputation:
1: **for** $i \in [m]$ **do**
2:   $L_{i,i} \leftarrow 0, R_{i,i} \leftarrow 0$
3:   **for** $j \in \{i+1,\dots,m\}$ **do** $L_{i,j} \leftarrow L_{i,j-1} + [c_i \succ_v c_j]$
4:   **for** $j \in \{i-1,\dots,1\}$ **do** $R_{j,i} \leftarrow R_{j+1,i} + [c_i \succ_v c_j]$
  Phase 2, Main Computation:
5: $A_{0,0} \leftarrow 0$
6: **for** $\ell \in [m-1]$ **do** $A_{\ell,0} \leftarrow A_{\ell-1,0} + L_{\ell,m}$
7: **for** $r \in [m-1]$ **do**
8:   $A_{0,r} \leftarrow A_{0,r-1} + R_{1,m+1-r}$
9:   **for** $\ell \in [m-r-1]$ **do**
10:     $A_{\ell,r} \leftarrow \min(A_{\ell-1,r} + L_{\ell,m-r},\ A_{\ell,r-1} + R_{\ell+1,m+1-r})$
11: **return** $\min_{\ell \in [m]} A_{\ell-1,m-\ell}$

As we will show, all values of $A_{\ell,r}$ can be computed efficiently in Algorithm 1, using a recursive formula.

Theorem 4.3. *Algorithm 1 computes the distance between a given vote and a single-peaked domain in time $O(m^2)$.*

Proof. For the running time, observe that each of our loops is over at most m elements, and we have at most two levels of nested loops. Each individual iteration can be completed in time $O(1)$. The final minimum in line 11 requires $O(m)$ time.

Let us now analyze the correctness of the algorithm. For each $i, j \in [m]$, with $i \leq j$, we let $L_{i,j}$ be the number of candidates in $\{c_i, c_{i+1}, \dots, c_j\}$ that v ranks below c_i. Consequently, we have that $L_{i,i} = 0$ and, if $i < j$, then either $L_{i,j} = L_{i,j-1} + 1$ (if v ranks c_i ahead of c_j) or $L_{i,j} = L_{i,j-1}$ (otherwise). Similarly, for $j \leq i$, $R_{i,j}$ is the number of candidates in $\{c_j, c_{j+1}, \dots, c_i\}$ that v ranks below c_i ($R_{j,i}$ satisfies analogous relations as $L_{i,j}$). The algorithm computes the values of $L_{i,j}$ and $R_{j,i}$ in Phase 1.

Then, in Phase 2, the algorithm computes all the values $A_{\ell,r}$ for $\ell, r \in [m]$ such that $\ell + r \leq m - 1$. Let us fix such ℓ and r. We note that every ranking in $U_{\ell,r}$ either ranks c_ℓ or c_{m+1-r} on position $m+1-\ell-r$ (i.e., on the $\ell+r$'th position from the bottom). Indeed, for all rankings in $U_{\ell,r}$ we have that the first $m+1-\ell-r$ candidates form an interval within $\rhd$. However, by definition, all of these candidates, except for the one ranked on position $m-\ell-r+1$, belong to $C \setminus C_{\ell,r}$. Consequently, to form the interval, the candidate on position $m-\ell-r+1$ must be either c_ℓ or c_{m+1-r}. Let $U_{\underline{\ell},r}$ be a subset of votes from $U_{\ell-1,r}$ that additionally have c_ℓ in the position $m+1-\ell-r$. Similarly, let $U_{\ell,\underline{r}}$, be a subset of votes from $U_{\ell,r-1}$ with c_{m+1-r} in the position $m+1-\ell-r$. By the preceding argument, we have that $U_{\ell,r} = U_{\underline{\ell},r} \cup U_{\ell,\underline{r}}$ (if $\ell = 0$, we assume $U_{\underline{\ell},r} = \emptyset$, if $r = 0$, $U_{\ell,\underline{r}} = \emptyset$). Thus, $A_{\ell,r} = \min(\mathrm{swap}(U_{\underline{\ell},r}, v), \mathrm{swap}(U_{\ell,\underline{r}}, v))$.

Let u be a vote in $U_{\underline{\ell},r}$ that minimizes $\mathrm{swap}(u, v)$. Observe that $m-\ell-r$ first candidates in u appear in the same relative order as they appear in v (otherwise, ordering them as in v would decrease the distance). Let u' be a vote obtained from u by ensuring that it ranks its first $m-\ell-r+1$ candidates in the same relative order as in v (in other words, u' is the same as u, except that it might rank c_ℓ some positions earlier). It must be that $u' \in U_{\ell-1,r}$. Moreover, we can show that u' minimizes swap distance to v among rankings in $U_{\ell-1,r}$, i.e., $\mathrm{swap}(u', v) = A_{\ell-1,r}$. Indeed, the first $m-\ell-r+1$ candidates are in the optimal order (the same as in v), and if rearranging the last $\ell+r-1$ candidates could decrease the distance, we could also rearrange them in the same way in u. Now, when we look at the inversions counted in $\mathrm{swap}(v, u)$, we see that we count all inversions that we count in $\mathrm{swap}(v, u')$ and additionally those from having c_ℓ after all of the first $m-\ell-r$ candidates. But those are exactly the inversions we store in $L_{\ell,m-r}$. Thus, we get that $\mathrm{swap}(U_{\underline{\ell},r}, v) = A_{\ell-1,r} + L_{\ell,m-r}$. Analogously, we can prove that $\mathrm{swap}(U_{\ell,\underline{r}}, v) = A_{\ell,r-1} + R_{\ell+1,m+1-r}$. This way, we obtain the recursive equation used in line 10, as well as the equations from lines 6 and 8 (in their cases either $r = 0$ or $\ell = 0$ so respective parts of the equation disappear).

Finally, for $\ell \in [m]$, we observe that $A_{\ell-1,m-\ell}$ is the minimal distance from v to a single-peaked ranking u in which c_ℓ is the top candidate. Thus, to get the overall smallest distance, we take the minimum from all these values. ◻

Every vote in SPOC is single-peaked along the axis obtained by "cutting" the cycle between some two adjacent candidates [30]. There are m such axes, hence we can run Algorithm 1 for each of them and choose the minimum distance. This gives as an algorithm running in time $O(m^3)$. We can improve that and get an $O(m^2)$ algorithm by a similar dynamic programming algorithm as for SP.

Theorem 4.4. *There is an algorithm that computes the swap distance between a given vote and* SPOC *in time $O(m^2)$.*

We can also extend Algorithm 1 to work for the case of SP(T), where T is an SP-tree. If T has k leaves (i.e., k nodes of degree 1), then the algorithm requires $O(km^k)$ time. The main idea is to implement dynamic programming over sets of connected vertices in T, of which there are $O(m^k)$.

Theorem 4.5. *There is an algorithm that given an* SP*-tree that has k leaves, computes the swap distance between a given vote and* SP(T) *in time $O(km^k)$.*

Given the algorithms for SP, SPOC, and single-peaked-on-a-tree domains, one could ask for a general polynomial-time algorithm that works for all single-peaked-on-a-graph domains. We prove that in this general case the problem is NP-complete.

Theorem 4.6. *Given a graph G, a vote $v \in \mathcal{L}(V(G))$, and an integer $d \in \mathbb{N}$, deciding if* $\mathrm{swap}(SP(G), v) \leq d$ *is* NP*-complete.*

4.2 Group-Separable Domains

For a group-separable domain with an arbitrary tree, we show an algorithm that computes the distance to a given vote in time $O(m^2)$.

Assume we are given a vote v and a group separable domain $D = \mathrm{GS}(T)$. Then, observe that finding a vote $u \in D$ that minimizes $\mathrm{swap}(u, v)$ is equivalent to reversing the order of some of the children of each internal node of T so that the frontier u of T minimizes $\mathrm{swap}(u, v)$. Moreover, the change in distance we get by reversing the order of the children of one particular node is independent of the configuration of the other nodes. Hence, we can consider internal nodes of tree T one by one, and for each decide in which of the two ways its children should be ordered. Fix such an arbitrary node with k children, and let $C_1, C_2, \dots, C_k$ denote the sets of candidates associated with leaves that are descendants of each of the children, when looking from left to right. This configuration would incur the distance of:

$$\sum_{1 \leq i < j \leq k} |\{(a, b) \in C_i \times C_j : b \succ_v a\}|,$$

while reversing the order gives the distance of:

$$\sum_{1 \leq i < j \leq k} |\{(a, b) \in C_i \times C_j : a \succ_v b\}|.$$

Thus, we compute the values of both sums and choose the configuration that leads to the lower one (or make an arbitrary choice in case of a tie). When considering all internal nodes of T in this way, we check each pair of candidates exactly once. Hence, the running time of this algorithm is $O(m^2)$.

Theorem 4.7. *There is an algorithm that given a* GS*-tree T and a vote v, computes* $\mathrm{swap}(\mathrm{GS}(T), v)$ *in time* $O(m^2)$.

For GS/bal and GS/cat, we give algorithms running in time $O(m \log m)$. Both algorithms follow the general approach outlined above, but for GS/bal we speed up computing inversions using an approach similar to that from the classic Merge Sort algorithm, and for GS/cat we use a special data structure.

Theorem 4.8. *There are algorithms that compute the swap distance between a given vote and* GS/cat *and* GS/bal *(represented via* GS*-trees) in time* $O(m \log m)$.

4.3 Single-Crossing and Euclidean Domains

Both single-crossing and Euclidean domains contain polynomially many votes, so a brute-force algorithm that given a ranking v computes its swap distance to all the rankings in the domain runs in polynomial time. For example, for SC, which contains $O(m^2)$ rankings, it would run in time $O(m^3\sqrt{\log m})$ [6]. However, as we typically want to compute the distance from many votes to our domains, we get better running times via appropriate preprocessing. Briefly put, for each domain $D \in$ {SC, 1D-Int., 2D-Square, 3D-Cube} we can arrange the rankings from these domains on a tree $T(D)$—or even on a path, in case of 1D-Int. and SC—so that two neighboring rankings are at swap distance one. Then, to compute a distance from a given ranking v to each member of the domain, we compute the distance between v and an arbitrary ranking in the domain, and then traverse the tree, updating the distance on the fly, so for each member of the domain we get its swap distance to v. Building $T(D)$ adds, at most, factor $O(m^2)$ to the complexity of computing the rankings from the domain.

Theorem 4.9. *For each D that is either* SC *or a Euclidean domain, there is an algorithm that given a ranking v and tree $T(D)$ computes* $\mathrm{swap}(D, v)$ *in time* $O(|D|)$.

5 ANALYSIS OF THE DOMAINS

Let us now analyze the outer diversity of our domains. We first consider the case of 8 candidates, and then we analyze how the outer diversities of our domains change as the number of candidates grows. The case of 8 candidates is interesting for the following, somewhat interrelated, reasons: (1) Faliszewski et al. [18] largely focused on this case, and we want our results to be comparable to theirs; (2) The case of 8 candidates is among the most popular ones in experiments within computational social choice [5]; (3) Considering only 8 candidates allows us to perform exact computations.

Table 2: Size, average normalized swap distance, outer diversity, and size of direct neighborhood (also normalized) of various domains, for the case of 8 candidates. The standard deviation of outer diversity for domains that we need to sample (SC, 1D-Int., 2D-Square, 3D-Cube) is no larger than 0.005 (for ten samples).

Domain D	$\|D\|$	ansd(D)	out-div(D)	dist-1	dist-1/$\|D\|$
Vote+Its Rev.	2	0.384	0.232	14	7
GS/cat	128	0.194	0.613	704	5.5
GS/bal	128	0.257	0.486	384	3
SP	128	0.284	0.432	384	3
SP/DF	496	0.239	0.522	968	1.952
SPOC	512	0.196	0.608	1280	2.5
SC	29	0.316	0.368	130.3	4.493
1D-Int.	29	0.311	0.378	134.8	4.648
2D-Square	351	0.217	0.566	988.0	2.815
3D-Cube	2311	0.138	0.724	3878.2	1.678
Largest Cond.	224	0.282	0.435	544	2.429

5.1 Outer-Diversity for Eight Candidates

For each of our domains, in Table 2 we provide its size, average normalized swap distance, outer diversity value, the number of votes in $\mathcal{L}(C)$ that are exactly at swap distance 1 from this domain (we refer to this as the *size of the direct neighborhood*), and the latter number normalized by the size of the domain (we analyze these values later on). Additionally, the table also includes LC domain, i.e., the largest Condorcet domain over 8 candidates, recently discovered by Leedham-Green et al. [27]. Sorting our domains with respect to their outer diversity values gives the following ranking:

$$\underset{0.719}{\text{3D-Cube}} \succ \{\underset{0.613}{\text{GS/cat}}, \underset{0.608}{\text{SPOC}}\} \succ \underset{0.565}{\text{2D-Square}} \succ \underset{0.522}{\text{SP/DF}} \succ \underset{0.486}{\text{GS/bal}} \succ \{\underset{0.435}{\text{LC}}, \underset{0.432}{\text{SP}}\} \succ \{\underset{0.386}{\text{1D-Int.}}, \underset{0.37}{\text{SC}}\}.$$

It is quite interesting that even though LC is the largest Condorcet domain over 8 candidates, its outer diversity is very similar to that of SP, which contains nearly half of the votes, and it is notably lower than outer diversities of GS/cat and GS/bal (both of the same cardinality as SP). However, a closer analysis of this domain confirms that it is not as diverse as one might expect given its size. For example, there are only 4 candidates that are ever ranked first in its votes, and 4 different candidate that are ever ranked last (indeed, the domain has further restrictions along these lines, which we omit due to limited space). Next, we note that our ranking is very similar to an analogous one obtained by Faliszewski et al. [18] based on inner diversity (also for the case of 8 candidates; note in their case there are no specific values measuring diversity and the ranking was obtained by comparing Kemeny vectors of the domains):

$$\text{GS/cat} \succ \text{3D-Cube} \succ \{\text{2D-Square, SPOC}\} \succ \{\text{SP/DF, GS/bal}\} \succ \text{SP} \succ \{\text{SC, 1D-Int.}\}.$$

Both rankings put 3D-Cube and GS/cat as the most diverse domains, and they both put 1D-Int. and SC as the least diverse ones. Further, they both rank domains from the same families identically: SPOC is more diverse than SP/DF, which is more diverse than SP, and

GS/cat is more diverse than GS/bal (not to mention the ranking of the Euclidean domains). The fact that 3D-Cube has higher outer diversity than GS/cat, as well as the tie between GS/cat and SPOC, are artifacts of considering only 8 candidates and for larger numbers of candidates these relations change (see Section 5.2).

Below, we analyze two features of our domains that are not directly related to capturing diversity, but which manifest themselves during outer diversity computations and which shed some light on how our domains are arranged within the general domain.

5.1.1 Direct Neighborhoods. The size of the direct neighborhood of a domain, normalized by the sizes of this domains, is interesting as it gives some intuition on how the domain is "spread" over $\mathcal{L}(C)$. For example, the domain that consists of a single ranking and its reverse is "maximally spread:" Its two members are as far apart as possible and, as we consider 8 candidates, there are exactly 7 rankings next to each of the domain members, neither of which belongs to the domain. Among our structured domains, GS/cat is the most spread one, with the value of 5.5, and 3D-Cube is the least spread, with the value of 1.678. Hence, members of 3D-Cube are packed quite closely within $\mathcal{L}(C)$. While one could think that this is a consequence of 3D-Cube's large size, $\mathcal{L}(C)$ contains more than 16 rankings for every ranking in 3D-Cube. It is interesting that for some domains the normalized sizes of their direct neighborhoods are appealing, round numbers (such as 3 for GS/bal or 5.5 for GS/cat). For GS/bal and GS/cat, we show that this is not a mere coincidence; for the other domains we leave this issue open.

Proposition 5.1. *Let D be the* GS/bal *domain for $m = 2^k$ candidates. For every ranking $v \in D$ there are exactly $2^{k-1} - 1$ unique ones from $\mathcal{L}(C) \setminus D$ at swap distance 1 from v.*

Proposition 5.2. *Consider* GS/cat *over $m \geq 4$ candidates. For every ranking $v \in$* GS/cat *there are exactly $m - 3$ unique ones from $\mathcal{L}(C) \setminus D$ at swap distance 1 from v, and one ranking from $\mathcal{L}(C)$ that is at swap distance 1 from v and one other ranking in* GS/cat.

5.1.2 Popularity. Given a domain $D \subseteq \mathcal{L}(C)$ and a ranking $v \in D$, we define its *popularity*, denoted $\mathrm{pop}(v)$, as the number of rankings from $\mathcal{L}(C)$ for which v is the closest member of D (if for a given ranking $u \in \mathcal{L}(C)$ there are p members of D that are closest to u, then u contributes $1/p$ to the popularity of each of them). The average popularity of a ranking in $|D|$ is equal to $|\mathcal{L}(C)|/|D|$ and by *normalized popularity* of a ranking v we mean the ratio between its popularity and this value. Namely, we have $\mathrm{npop}(v) = \frac{\mathrm{pop}(v)}{|L(C)|/|D|}$. Popularity gives hints on both the internal symmetry of a domain, and the arrangement of its rankings in $\mathcal{L}(C)$. Indeed, the more uniform are the popularity values of the rankings, the more likely it is that they are symmetrically spread within $\mathcal{L}(C)$. On the other hand, a mixture of high and low popularity values suggests that the more popular rankings are on the "outskirts" of the domain, and the less popular ones belong to its "interior." We show the normalized popularities of the rankings in our domains in Figure 1, on the microscope plots of Faliszewski et al. [14].

Remark 5.1. *Let D be a domain. A microscope plot of D presents each ranking from the domain as a dot, whose Euclidean distance from the other dots is as similar to the swap distance between the respective rankings as possible (exact correspondence between Euclidean distances and swap distances is, typically, impossible to achieve, but microscopes still give useful intuitions).*

The plots show some remarkable features of our domains. The first observation is that for both GS/bal and GS/cat, all rankings have equal popularity, equal to the expected one. Indeed, this is a general feature of group separable domains.

Proposition 5.3. *Let $D = \mathrm{GS}(T)$ be a group separable domain over candidate set C. Then, for each $v \in \mathrm{GS}(T)$, $\mathrm{npop}(v) = 1$.*

The other domains show a high variance in popularity among their members. For example, the most popular rankings in SP are the societal axis and its reverse, whereas most rankings in between these two have low popularity. Overall, group-separable domains are perfectly symmetric and clearly stand out.

5.2 Outer Diversity for Larger Candidate Sets

When considering more than eight candidates, we compute outer diversity using the sampling approach, with sample size $N = 1000$ (see Section 4). For each domain, we repeat this computation 10 times, to also obtain standard deviation (it is so small as to be nearly invisible on our plots, which justifies the use of sampling).

In Figure 3, we show how the outer diversity of our domains evolves as a function of the number m of candidates, for $m \in \{2, 3, \dots, 20\}$. In particular, we note that the outer diversity of polynomially-sized Euclidean domains drops much more rapidly than that of the other, exponential-sized, ones. It is also notable how SPOC becomes less diverse than GS/cat (for 9 candidates or more) and how GS/cat becomes the most diverse among our domains (for 12 candidates or more). Further, GS/cat is consistently more diverse than GS/bal. As these two domains are extreme among the group-separable ones (one uses the tallest binary GS-tree and the other one the shortest), we ask if GS/cat is the most diverse group-separable domain and GS/bal is the least diverse one.

It is interesting if outer diversity of our domains eventually approaches zero, or if it stays bounded away from it. As shown below, the former happens, e.g., if the size of the domain is bounded by a constant, whereas the latter happens, e.g., for GS/cat. Hence, outer diversity of a domain may be bounded away from zero even if its size grows notably more slowly than that of the general domain (as a function of the number of candidates).

Proposition 5.4. *Let us fix value k and let $D_2, D_3, \dots$ be a sequence of domains, where each D_m contains at most k rankings over m candidates. Then $\lim_{m\to\infty} \text{out-div}(D_m) = 0$.*

Proposition 5.5. *If the number of candidates is even, then:*

$$\text{out-div}(\mathrm{GS/cat}) > 1/2.$$

Experimentally, we checked that for 1000 candidates, outer diversity of SP and SPOC is equal to around 0.039 and 0.055, respectively, raising a question of convergence to zero for these domains as well.

6 MOST DIVERSE DOMAINS

Given a number k, we ask for a domain of k rankings with the highest outer diversity value. As per our observation in Section 3, we can compute such a domain by solving the k-Kemeny problem

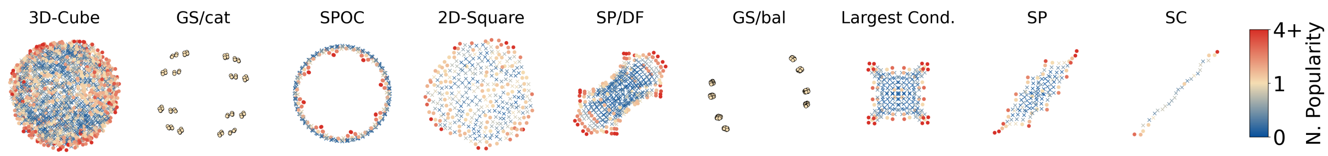

Figure 1: Microscope plots of our domains, where each dot/cross represents a ranking from the domain, colored according to its normalized popularity (see Remark 5.1). Rankings with normalized popularity below 1 are marked with crosses, and the remaining ones with dots. Dots marking rankings with normalized popularity equal to exactly 1 have a black border.

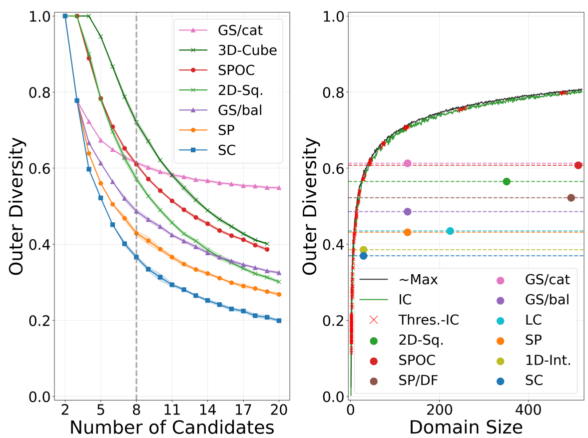

Figure 2: Outer diversity of several structured domains as a function of the number of candidates (on the left), or as a function of their size (on the right; including approximations of most diverse domains). For SPOC and 3D-Cube, we omit outer diversity for 20 candidates, due to computation time.

for the UN election using, e.g., integer linear programming (ILP).[2] Unfortunately, solving this ILP is challenging, as its size for m candidates is $\Theta((m!)^2)$. Hence, for $m \geq 6$ we use the following heuristics (to compute the outer diversity of the domains produced by them, we use the sampling approach, with $N = 1000$ samples):

(1) We sample k rankings uniformly at random from $\mathcal{L}(C)$ (this is known as sampling from impartial culture, IC).
(2) We sample k rankings from IC and perform simulated annealing (technical details available in the full version).

We also use a heuristic that does not allow us to control the size of the domain, but selects rankings that are spread out over $\mathcal{L}(C)$:

(3) We choose a threshold $t \in \{5, 6, \ldots, 25\}$ and iteratively sample 10^4 rankings from IC. We keep a ranking only if its swap distance to a closest already-kept one is greater or equal to t.

Instead of using this heuristic, we would rather keep on selecting rankings that are at the largest possible swap distance from those previously selected, but finding such rankings is NP-complete.

Theorem 6.1. *Given a positive integer t and a domain $D \subseteq \mathcal{L}(C)$, represented by explicitly listing its rankings, deciding if there is a ranking v such that $\min_{u \in D} \mathrm{swap}(u, v) \geq t$ is NP-complete.*

[2] In essence, this is k-Median clustering applied on the metric space of all possible rankings under the swap distance. We use its standard ILP formulation.

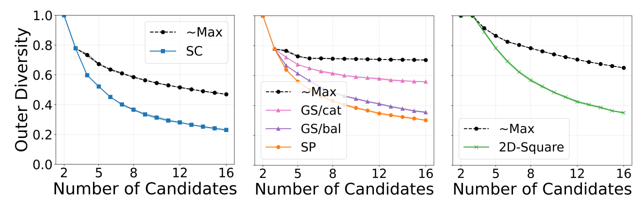

Figure 3: Outer diversity of structured domains as a function of the number of candidates, compared to that of (an approximation of) the most diverse domain of the same size.

On the plots, we denote domains computed using the first heuristic as IC, those computed using simulated annealing as ~Max, and those using the threshold approach as Thres.-IC. In Figure 2 (right) we show the outer diversities of these domains for the case of $m = 8$ candidates, as we increase k (for the first two heuristics) or decrease t (for the third one). We see that all three heuristics produce very similar results. We interpret this as suggesting that, indeed, we get close to the highest possible diversities. For the case of 6 candidates we also compared our heuristically computed domains to the optimal ones, obtained using ILP, and the results were nearly identical. Figure 2 (right) also includes points corresponding to our structured domains, illustrating how far off they are from the most diverse domains of their size.

In Figure 3, for each domain $D \in \{\mathrm{SC}, \mathrm{GS/cat}, \mathrm{GS/bal}, \mathrm{SP}, \text{2D-Square}\}$, we plot the outer diversity of this domain and the outer diversity of the most diverse domain of size $|D|$ (as computed using our second heuristic) as a function of the number of candidates. In particular, we see that for polynomial-sized domains (SC and 2D-Square), the diversity of the most diverse domains seems to be dropping up to 16 candidates. In contrast, for SP, GS/bal, and GS/cat, which are all of size 2^{m-1}, the outer diversity of the most diverse domain seems to stabilize around the value 0.7.

7 CONCLUSIONS

Our main conclusion is that outer diversity is a useful, practical measure of domain diversity. Using it, we have found that GS/cat sharply stands out from many other structured domains in various respects and, so, we recommend its use in experiments. Throughout the paper, we have made a number of observations, and we have explained some of them theoretically. We propose seeking such explanations for the remaining observations as future work. Another interesting direction for future work is to establish a formal relation between outer and inner diversity notions.

ACKNOWLEDGMENTS

Tomasz Wąs was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant EP/X038548/1. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101002854). This work was funded with a grant from the Foundation for the University of Geneva.

REFERENCES

[1] M. Ammann and C. Puppe. 2025. Preference Diversity. *Review of Economic Design* (2025). Online First.

[2] J. Bartholdi, III, C. Tovey, and M. Trick. 1989. Voting Schemes for Which it Can Be Difficult to Tell Who Won The Election. *Social Choice and Welfare* 6, 2 (1989), 157–165.

[3] T. Biedl, F. J. Brandenburg, and X. Deng. 2009. On the Complexity of Crossings in Permutations. *Discrete Mathematics* 309, 7 (2009), 1813–1823.

[4] D. Black. 1958. *The Theory of Committees and Elections.* Cambridge University Press.

[5] N. Boehmer, P. Faliszewski, L. Janeczko, A. Kaczmarczyk, G. Lisowski, G. Pierczyński, S. Rey, D. Stolicki, S. Szufa, and T. Wąs. 2024. Guide to Numerical Experiments on Elections in Computational Social Choice. In *Proceedings of IJCAI-2024.* 7962–7970.

[6] T. Chan and M. Pătraşcu. 2010. Counting Inversions, Offline Orthogonal Range Counting, and Related Problems. In *Proceedings of SODA-10.* 161–173.

[7] G. Demange. 1982. Single-Peaked Orders on a Tree. *Mathematical Social Sciences* 3, 4 (1982), 389–396.

[8] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. 2001. Rank Aggregation Methods for the Web. In *Proceedings of WWW-01.* 613–622.

[9] E. Elkind, P. Faliszewski, and A. Slinko. 2015. Distance Rationalization of Voting Rules. *Social Choice and Welfare* 45, 2 (2015), 345–377.

[10] E. Elkind, M. Lackner, and D. Peters. 2017. Structured Preferences. In *Trends in Computational Social Choice*, U. Endriss (Ed.). AI Access Foundation, 187–207.

[11] J. Enelow and M. Hinich. 1984. *The Spatial Theory of Voting: An Introduction.* Cambridge University Press.

[12] J. Enelow and M. Hinich. 1990. *Advances in the Spatial Theory of Voting.* Cambridge University Press.

[13] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. 2014. The Complexity of Manipulative Attacks in Nearly Single-Peaked Electorates. *Artificial Intelligence* 207 (2014), 69–99.

[14] P. Faliszewski, A. Kaczmarczyk, K. Sornat, S. Szufa, and T. Wąs. 2023. Diversity, Agreement, and Polarization in Elections. In *Proceedings of IJCAI-2023.* 2684–2692.

[15] P. Faliszewski, A. Karpov, and S. Obraztsova. 2022. The complexity of election problems with group-separable preferences. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 18.

[16] P. Faliszewski, J. Mertlová, P. Nunn, S. Szufa, and T. Wąs. 2025. Distances Between Top-Truncated Elections of Different Sizes. In *Proceedings of AAAI-2025.* 13823–13830.

[17] P. Faliszewski, P. Skowron, A. Slinko, K. Sornat, S. Szufa, and N. Talmon. 2025. How Similar Are Two Elections? *J. Comput. System Sci.* 150 (2025), 103632.

[18] P. Faliszewski, K. Sornat, S. Szufa, and T. Wąs. 2025. *Diversity of Structured Domains via k-Kemeny Scores.* Technical Report arXiv:2509.15812 [cs.GT]. arXiv.org.

[19] P. Faliszewski, K. Sornat, S. Szufa, and T. Wąs. 2026. *Outer Diversity of Structured Domains.* Technical Report arXiv:2602.15708 [cs.GT]. arXiv.org.

[20] V. Hashemi and U. Endriss. 2014. Measuring Diversity of Preferences in a Group. In *Proceedings of ECAI-2014.* 423–428.

[21] E. Hemaspaandra, H. Spakowski, and J. Vogel. 2005. The Complexity of Kemeny Elections. *Theoretical Computer Science* 349, 3 (2005), 382–391.

[22] K. Inada. 1964. A Note on the Simple Majority Decision Rule. *Econometrica* 32, 32 (1964), 525–531.

[23] K. Inada. 1969. The Simple Majority Decision Rule. *Econometrica* 37, 3 (1969), 490–506.

[24] A. Karpov. 2019. On the number of group-separable preference profiles. *Group Decision and Negotiation* 28, 3 (2019), 501–517.

[25] A. Karpov, K. Markström, S. Riis, and B. Zhou. 2024. *Local Diversity of Condorcet Domains.* Technical Report arXiv:2401.11912 [econ.TH]. arXiv.org.

[26] J. Kemeny. 1959. Mathematics Without Numbers. *Daedalus* 88 (1959), 577–591.

[27] C. Leedham-Green, K. Markström, and S. Riis. 2024. The Largest Condorcet Domain on 8 Alternatives. *Social Choice and Welfare* 62, 1 (2024), 109–116.

[28] J. Mirrlees. 1971. An Exploration in the Theory of Optimal Income Taxation. *Review of Economic Studies* 38 (1971), 175–208.

[29] K. Nehring and C. Puppe. 2002. A Theory of Diversity. *Econometrica* 70, 3 (2002), 1155–1198.

[30] D. Peters and M. Lackner. 2020. Preferences Single-Peaked on a Circle. *Journal of Artificial Intelligence Research* 68 (2020), 463–502.

[31] K. Roberts. 1977. Voting Over Income Tax Schedules. *Journal of Public Economics* 8, 3 (1977), 329–340.

[32] S. Szufa, N. Boehmer, R. Bredereck, P. Faliszewski, R. Niedermeier, P. Skowron, A. Slinko, and N. Talmon. 2025. Drawing a map of elections. *Artificial Intelligence* 343 (2025), 104332.