Discovering Consistent Subelections

Łukasz Janeczko
AGH University
Kraków, Poland
ljaneczk@agh.edu.pl

Jérôme Lang
CNRS, LAMSADE, Université Paris Dauphine-PSL
Paris, France
lang@lamsade.dauphine.fr

Grzegorz Lisowski
AGH University
Kraków, Poland
glisowski@agh.edu.pl

Stanisław Szufa
AGH University, Poland
CNRS, LAMSADE, Université Paris Dauphine-PSL, France
s.szufa@gmail.com

ABSTRACT

We show how hidden interesting subelections can be discovered in ordinal elections. An interesting subelection consists of a reasonably large set of voters and a reasonably large set of candidates such that the former have a consistent opinion about the latter. Consistency may take various forms but we focus on three: Identity (all selected voters rank all selected candidates the same way), antagonism (half of the selected voters rank candidates in some order and the other half in the reverse order), and clones (all selected voters rank all selected candidates contiguously in the original election). We first study the computation of such hidden subelections. Second, we analyze synthetic and real-life data, and find that identifying hidden consistent subelections allows us to uncover some relevant concepts.

KEYWORDS

Subelections; Ordinal Elections; Clones; Computational Complexity

ACM Reference Format:
Łukasz Janeczko, Jérôme Lang, Grzegorz Lisowski, and Stanisław Szufa. 2024. Discovering Consistent Subelections. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

Ordinal voting consists in taking as input a preference profile (a collection of rankings over candidates) and producing a winner, or a set of winners, or a collective ranking as output — this could be called the "mechanism" view of voting. A much less studied question consists in discovering, from a preference profile, some hidden properties of the domain at hand. An example of such a study is the discovery of structure among the set of candidates, such as an order of candidates that makes the profile single-peaked (perhaps only approximately), or among the set of voters, such as an order that makes the profile single-crossing.

There is however more to discover from a preference profile, by focusing on voters and candidates *simultaneously.* Imagine an explorer from another planet visiting us and observing a local voting profile over food items. They do not know anything about our food, they do not have the concepts of meat, fish, vegetables, sweet, or spicy. They do not have either the concepts of children, or vegetarians, and ignore our local cultures. Still, they can observe that a significant group of voters consistently prefer some items to others (say, tofu and lentils to eggs, eggs to fish, and fish to meat) and that a significant group of voters are indifferent to all food items that they do not know as it is not part of their culture.

In a more realistic context, the voters are citizens of a country with unknown world views, and the candidates are political issues. Still in another context, the explorer is a manufacturing company, voters are potential consumers, and candidates are items they would be interested to purchase if they were on the market.

The hidden information we seek is a subset of voters V' and a subset of candidates C' such that voters in V' have *consistent* preferences over V', i.e., it is a *consistent subelection* of the original election. In order for a subelection to make us learn something interesting, the consistency property has to be meaningful. We focus on the following three consistency properties:

Identity: All the candidates are ranked the same way by all voters (children prefer coke to juice and juice to coffee).

Clone structure : Voters in V' rank all candidates in C' contiguously in the original election (people living outside of Europe are indifferent between non-exported varieties of European cheese; this is not to say they rank them at the bottom, as they may very well rank them above items that they know they do not like).

Antagonism : Half of the voters in V' rank the candidates in C' in the same order, while the others rank them in the opposite one.

Identity is probably the most interesting consistency property. It allows us to discover significant segments of the population with identical preferences on a large fraction of options (e.g., it allows us to discover that a homogeneous set of voters, whatever we want to call it, prefers tofu to eggs, eggs to fish, and fish to meat). Clone structures also lead to major findings: A subpopulation considering a set of alternatives as clones usually means that they cannot distinguish between them (e.g., uncommon cheese varieties). We include antagonism as it is the natural opposite of identity and it helps us to discover what divides a subpopulation. We leave other meaningful properties, such as single-peakedness, single-crossingness or group separability for further research.

Note an important difference between, on the one hand, identity and antagonism, and on the other, clone structures: We do not need the original election to check if a subelection satisfies identity or

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

single-peakedness, whereas being a clone structure can only be defined with respect to the original election, which is not the case for identity or antagonism.

In order for a consistent subelection to be meaningful, not only should the property make sense, but the size of the subelection should also be reasonably large (for instance, knowing that two voters out of ten are consistent over three candidates out of ten does not tell us anything interesting). This motivates searching for consistent subelections whose number of voters (resp. candidates) reach a given threshold.

Example 1. *To clarify the concepts of hidden clones, identity, and antagonism, consider the following election, with six voters expressing preferences over six candidates:*

$$\begin{aligned} v_1 &: a \succ b \succ c \succ f \succ e \succ d \\ v_2 &: c \succ b \succ a \succ d \succ e \succ f \\ v_3 &: a \succ f \succ e \succ b \succ c \succ d \\ v_4 &: d \succ e \succ f \succ c \succ b \succ a \\ v_5 &: a \succ c \succ b \succ d \succ e \succ f \\ v_6 &: f \succ e \succ a \succ b \succ c \succ d \end{aligned}$$

We discover some interesting patterns:

- *All voters agree that candidates e and f are clones; and all voters except v_3 agree that a, b, and c are clones: We have a clone set of two candidates for six voters, and a clone set of three candidates for five voters.*
- *v_1, v_3 and v_6 agree on the ranking $a \succ b \succ c \succ d$: This is a hidden identity with three voters and four candidates. All voters except v_2 and v_4 agree on $a \succ b \succ d$ and on $a \succ c \succ d$: These are hidden identities with four voters and three candidates.*
- *Candidates d, e, and f are antagonizing all voters: Half of them (v_2, v_4, and v_5) rank them $d \succ e \succ f$, while the other half (v_1, v_3, and v_6) rank them in the reverse order $f \succ e \succ d$: There is an antagonism for three candidates and six voters.*

Discovering consistent, large enough subelections not only helps to understand a population better (*e.g.*, with respect to food preferences, customer behavior, or political opinions over issues), but it can also benefit a variety of tasks. Notably, if we start eliciting the preferences of a new voter (outside of the original profile), we might discover that they probably belong to some group with consistent preferences, which eases and speeds up the elicitation process. For instance, once discovered that Ann is a vegetarian, asking her if she prefers pork to asparagus is a loss of time.

For the sake of simplicity, in this paper, we choose to focus on subelections that satisfy *exact* consistency properties. Occasionally, if applicable, we relate to the question of finding the closest subelection in terms of swap distance, that is, the minimum number of swaps on adjacent candidates we need to perform to obtain a subelection satisfying some consistency property [10]. While allowing approximate properties (e.g., find a subelection where all voters order almost all candidates in the same way) is definitely interesting, and of course, would allow us to discover larger meaningful subelections, we leave it for further research.

Our Contribution. We provide an analysis of finding hidden subelections, both from theoretical and empirical perspectives. First, we focus on their computation. We obtain hardness results for finding a sufficiently large hidden identity or antagonism, which are tempered by parameterized tractability results for the number of candidates or voters, and by a translation into an ILP. We note that we obtain a reduced run time of our FPT algorithms due to a graph representation of unanimous preference orders. As to hidden clones, they can be identified in polynomial time.

Then, we perform an empirical analysis, using both synthetic and real-life data. To analyze the results for synthetic data we use the *map of elections* framework. As to the real-life data, we study a well-known dataset containing preferences over different types of sushi, as well as a political election dataset (obtained through polls) over candidates from the 2014 French presidential election.

Related Literature. Discovering structure in elections has received significant attention in the context of *single-peakedness*: Given a profile, is it single-peaked with respect to some hidden axis (such as political left-right) on which the candidates are positioned? Since the plausibility of a positive answer quickly decreases with the size of the profile, most of the focus has been laid on approximate single-peakedness; several measures of single-peakedness have been defined, and the key question is to identify an axis for which the profile maximizes the single-peakedness degree. Variants have been considered (such as Euclidean preferences, single-peakedness on a tree or a circle), as well as other structures (mostly single-crossingness, where the hidden axis bears on the set of voters). The most recent review of work on this trend is [9].

Uncovering a hidden axis and maximally explaining the preference profile allows us to discover hidden properties of the domain at hand, which is also our motivation. However, uncovering an axis allows us to learn structure *of the set of candidates only.* Symmetrically, finding an axis over the set of voters making the profile as much *single-crossing* as possible leads us to learning structure *of the set of voters only.*

Elkind et al. [8] identify *clones* in elections (sets of candidates that are ranked contiguously by all voters). Part of our contribution generalizes the discovery of clone sets by considering sets of candidates that are considered clones by only some of the voters.[1]

Colley et al. [6] identify, from a preference profile, the *divisiveness* of each candidate, measuring the disagreement of the population about it. Various notions of degrees of consensus, conflict, or diversity within an electorate have also been studied [1, 2, 15]. While (some of) the subelections we discover also tell us something about the degree of consensus or conflict in a society, they tell us much more, by localizing the candidates *and* the voters on which there is a high consensus or conflict.

Faliszewski et al. [12] study the problem of subelection isomorphism, where they analyze the complexity of verifying whether one election is isomorphic to a subelection of the other election. This approach differs from ours because we always focus solely on the inner structure of a single election, hence, we do not have to

[1] A specific notion of clone structure occurs when the set of candidates can be partitioned in two classes, such that each voter prefers all candidates in one class to all those in the other one (*group separability*); see Sections 3.11 and 4.7 of (Elkind, Lackner, and Peters 2022).

deal with all the problems related to matching the candidates and voters from different elections.

Biclustering [14], also known as co-clustering or block-clustering, aims at learning structure in a real-valued, two-dimensional matrix by simultaneously finding a set of rows and a set of columns with similar behavior, i.e., near-identical rows, near-identical columns, rows or columns roughly obtained from each other by an additive or multiplicative factor. This resembles our subelection discovery tasks, with a major difference: Biclustering algorithms work on a cardinal input, while ours is ordinal. This is more important than it may appear. Crucially, expressing a profile as a matrix whose cell corresponding to voter i and candidate c_j is c_j's rank in i's ranking would not help, as our consistency notions cannot be expressed by near-identity or near-linear relations between rows or columns.

2 CONSISTENT SUBELECTIONS

Let us introduce the basic notions which we use in our analysis. For a natural number $t > 0$, we denote as $[t]$ the set $\{1, \dots, t\}$.

Elections. An *election* $E = (C, V)$ consists of the set C of *candidates* and the set V of *voters*. We assume that each voter v submits a *ranking* $\succ_v$ over C. Also, The *size* of an election E, denoted $size(E)$, is a pair of integers $(|C|, |V|)$.

Subelections. For an election $E = (C, V)$, a subset of candidates $C' \subseteq C$, and a subset of voters $V' \subseteq V$, we say that $E' = (C', V')$ is a *subelection* of E for every voter $v \in V'$ and any pair of candidates $c_1, c_2 \in C'$, $c_1 \succ_v c_2$ in E' if and only if $c_1 \succ_v c_2$ in E.

Identity. We say that $E = (C, V)$ is an *identity election* if, for every pair of voters $v_i, v_j \in V$, $\succ_{v_i} = \succ_{v_j}$.

Antagonism. We say that $E = (C, V)$, with $|V|$ even, is an *antagonism* if there is a partition of V into V_1 and V_2 such that $|V_1| = |V_2|$ and for every pair of voters $v_1 \in V_1$, $v_2 \in V_2$, as well as any pair of candidates $c, c' \in C$, $c \succ_{v_1} c'$ if and only if $c' \succ_{v_2} c$.

Clones. We say that C' is a clone set for an election $E = (C, V)$ if for any $x, y \in C'$, any $z \notin C'$ and any voter i we have $x \succ_i z$ if and only if $y \succ_i z$. The *size* of a clone set C' for election E is the pair of integers $(|C'|, |V|)$.

In our initial example, $(\{a, b, c, d\}, \{v_1, v_3, v_6\})$ is an identity subelection of E of size $(4, 3)$; $(\{a, b, c\}, V)$ is an antagonism subelection of E of size $(3, 6)$; and $\{a, b, c\}$ is a clone set for the subelection $(C, \{v_1, v_2, v_4, v_5, v_6\})$ of E of size $(3, 5)$.

Given two identity (resp. antagonism) subelections E_1, E_2 for election E, we say that E_1 is larger than E_2 if $size(E_1) = (m_1, n_1)$, $size(E_2) = (m_2, n_2)$, $m_1 \geq m_2$, and $n_1 \geq n_2$, with one of these two inequalities being strict. In other terms, this is Pareto-dominance for two criteria, being the number of candidates and the number of voters in subelections. Note also that the existence of an identity of size m' for n' voters implies the existence of an identity of size $m'' \leq m'$ for $n'' \leq n'$ voters.

We define the *identity* (resp. *antagonism*) signature of an election as the set of pairs (m', n') of integers such that (i) there is an identity (resp. antagonism) subelection E' of E of size (m', n'), and (ii) no identity (resp. antagonism) subelection of E is larger than E'. In Example 1, the identity signature of E is $\{(1, 6), (3, 4), (4, 3), (6, 1)\}$. We will further say that a set of voters (resp. candidates) has an identity (resp. antagonism) subelection if there is a subelection of that type with that set of voters or candidates.

We assume that the reader is familiar with basic concepts in (parametrized) computational complexity.

3 COMPUTING MEANINGFUL SUBELECTIONS

In this section, we study the complexity of discovering hidden subelections and provide algorithms for finding them.

3.1 Hidden Clones

Let us commence with the problem of finding clone sets in an election. In Hidden-Clones we are concerned with checking the existence of sufficiently large sets of voters similar with respect to a set of candidates of a given size.

Hidden-Clones:
Input: Election $E = (C, V)$, $m', n' \in \mathbb{N}$.
Question: Is there a set of n' voters $V' \subseteq V$ and a set of m' candidates $C' \subseteq C$ such that C' is a clone set for V'?

Theorem 1. *Hidden-Clones is P-time solvable.*

Proof. Let (E, m', n') be an instance of Hidden-Clones. A subset of m' candidates can be a clone set for at least one voter if and only if it forms a segment of m' consecutive candidates in some vote. We observe that there are at most $|V| \cdot (|C| - m' + 1)$ such subsets, we iterate over them and accept if any of them appear in at least n' votes. □

To better understand the algorithm, let us analyze it in the previous example.

Example 2. *Consider again the election from Example 1. Take $m' = 3$: The segments of length 3 for voter 1 are $\{a, b, c\}$, $\{b, c, f\}$, $\{c, e, f\}$ and $\{d, e, f\}$; for voter 2, these are $\{a, b, c\}$, $\{a, b, d\}$, $\{a, d, e\}$ and $\{d, e, f\}$; and so on. Counting the number of voters for which these subsets correspond to a segment of length 3, we find that $\{a, b, c\}$ appears four times, $\{b, c, f\}$ only once, $\{c, e, f\}$ twice, and so on. The sets occurring most frequently are $\{a, b, c\}$, $\{d, e, f\}$ (four times each): In conclusion, there are hidden clones of size $(3, n')$, for $n' \leq 3$.*

The following observation states that the existence of clone sets of a given size is not monotonic (whereas for identity and antagonism subelections, monotonicity holds.)

Observation 1. *The existence of a clone set of size m' does not imply the existence of a clone set of smaller (nor larger) size.*

As a trivial example, $(a \succ b \succ c, b \succ c \succ a, c \succ a \succ b)$ has no clone set of two candidates, but every singleton, as well as the set of all candidates, are clone sets for all voters.

For this reason, we do not define the clone signature of an election, but for every $m' \leq m$ we define MaxClone(E, m') as the largest $n' \in [n]$ such that there exists a set of n' voters $V' \subseteq V$ and a set of m' candidates $C' \subseteq C$ such that C' is a clone set.

Example 3. *Let us consider the instance defined in Example 1. There, we observe that MaxClone$(E, 2) = 6$, MaxClone$(E, 3) = 5$, MaxClone$(E, 4) =$ MaxClone$(E, 5) = 3$, and trivially, MaxClone$(E, 1) =$ MaxClone$(E, 6) = 6$.*

Analogously to the algorithm described in Theorem 1, we can also compute MaxClone in polynomial time (we search for a subset with maximum occurrences).

Corollary 2. *MaxClone is P-time solvable.*

We note that using standard data structures such as hash map and doubly linked list, both of our algorithms can be implemented in $O(|V| \cdot (|C| - m') \cdot m')$ time and space complexity, which makes them very fast and usable in practice. For example, for elections with a few hundred voters and a few hundred candidates, the algorithm works in a few seconds.

Due to the nonmonotonicity of clone sets, it is meaningless to approximate the size of a maximal clone set. Regarding the approximation of the maximal number of voters for which there exists a clone set of a given size, solving such a problem is very similar to solving the MaxClone problem.

We may also be interested in how far we are from obtaining a clone set of a given size for a given number of voters in terms of swap distance. However, the closest clone set (in terms of swap distance) may not be present in any vote: for $C = \{a, b, c, d, e, f, g, h, i\}$, $v_1 = \{a \succ b \succ c \succ d \succ e \succ f \succ g \succ h \succ i\}, v_2 = \{d \succ f \succ g \succ i \succ a \succ b \succ c \succ h \succ e\}, v_3 = \{a \succ b \succ c \succ f \succ e \succ i \succ d \succ h \succ g\}$, $n' = 3, m' = 4$, the closest clone set is $\{a, b, c, e\}$ with swap distance 3. However, for any clone set candidate cs, we can compute in polynomial time the minimum number of swaps we need to do so that cs becomes a valid clone set (and use it to find the best one).

3.2 Hidden Identity

We study the problem of whether a certain number of voters agree regarding the order of a given number of candidates.

Hidden-ID:
Input: Election $E = (C, V), m', n' \in \mathbb{N}$.
Question: Is there an identity subelection $E' = (C', V')$ of E, with $|C'| \geq m'$ and $|V'| \geq n'$?

We first show that Hidden-ID is intractable.

Theorem 3. *Hidden-ID is NP-complete.*

Proof Sketch. Membership to NP is clear as we can guess both a set of voters and a set of candidates and check if these voters rank these candidates in the same order. Hardness is shown by reduction from 3-SAT. For a 3-CNF formula ϕ with the set of variables $X = \{x_0, \dots, x_n\}$ and the set of clauses $C = \{C_0, \dots, C_m\}$, we consider a sufficiently large number of pairs of voters M, whom we call *main voters*. We further include three *clause voters* for each of the clauses in C. We will associate every such voter with a distinct literal in a corresponding clause. Subsequently, we consider a set of variables X', consisting of X and five additional variables for each clause, assuming an order of candidates in X'. Then, we construct a pair of *literal candidates* for each such variable x_i, i.e., c_{x_i} and $c_{\neg x_i}$. Furthermore, in the constructed instance of Hidden-ID we take the identity with the number of candidates equal to $|X'|$, and the number of voters $|C| + M$.

Further, we let main voters rank the candidates following the order of X', with one of such voters in each pair ranking $c_{x_i} \succ c_{\neg x_i}$ and the other $c_{\neg x_i} \succ c_{x_i}$, for every $x_i \in X'$. Notice that by choosing a sufficiently high number of main voters we ensure that in a subelection required in the instance we consider, we select exactly one literal candidate for each variable. Hence, such a subelection corresponds to some valuation over X'. We let every clause voter rank the candidate encoding the negation of a literal that the voter corresponds to lower than the candidates representing a variable with a higher index. Observe that then, if a clause voter v corresponding to a literal L is selected in a target subelection, then $c_{\neg L}$ is not, as then v's vote would not be identical to the main voters' rankings. Subsequently, using additional variables in X' corresponding to a clause C_i, we ensure that at most one of the clause voters for C_i is present in a subelection satisfying criteria of our instance, because otherwise the selected clause voters would not have identical votes in the target subelections. As by the size of a target subelection we need to select at least $|C|$ clause voters, we obtain that it exists exactly when φ is satisfiable. (See the full version of the paper for an example of votes in an encoding we define). □

Following the NP-hardness of the problem we consider in general, a natural approach is to ask for FPT algorithms. We will show that we are able to efficiently verify if a given set of voters has an identity of a given size (even if we do not know which candidates should be selected). This enables us to provide an FPT algorithm parameterized by the number of voters.

Proposition 4. *Checking if for a given set of n' voters there exists an identity subelection with at least m' candidates is P-time solvable.*

Proof. Suppose we are given a Hidden-ID instance (E, m', n') and a set of n' voters $V' = \{v_{i_1}, v_{i_2}, \dots, v_{i_{n'}}\}$. We ask if there exist m' candidates that are preferred exactly in the same order by all voters from V'.[2] We define the *unanimity graph $Una(V')$* as the graph whose set of vertices is C, and that contains edge (c, c') if and only if all voters in V' prefer c to c'. $Una(V')$ is a directed acyclic graph, and (C', V') is an identity subelection of E if and only if there is a path in $Una(V')$ that goes through all candidates of C'. Therefore, it suffices to check whether $Una(V')$ contains a path of length $m' - 1$; as finding the longest path in DAG is well-known to be P-time solvable [7], our algorithm runs in polynomial time. □

Specifically, the Algorithm 4 can be implemented in $O(n' \cdot |C|^2)$ time complexity and $O(n' \cdot |C| + |C|^2)$ space complexity. We demonstrate our approach in an example.

Example 4. *We continue Example 1. Take $V' = \{v_1, v_3, v_6\}$: The unanimity graph for V' consists of the edges $a \longrightarrow b$, $a \longrightarrow c$, $a \longrightarrow d$, $a \longrightarrow e$, $b \longrightarrow c$, $b \longrightarrow d$, $c \longrightarrow d$, $e \longrightarrow d$, $f \longrightarrow d$, $f \longrightarrow e$. The longest path being $a \longrightarrow b \longrightarrow c \longrightarrow d$, V' has a subelection with* 4 *candidates (and no more). With $V'' = \{v_1, v_2, v_3, v_6\}$, the unanimity graph is composed of the edges $a \longrightarrow d$, $b \longrightarrow c$, $b \longrightarrow d$, the longest path is $b \longrightarrow c \longrightarrow d$, V'' has a subelection with* 3 *candidates (and no more).*

Using the above algorithm, we obtainthat Hidden-ID is fixed-parameter tractable for the number of voters.

Corollary 5. *Hidden-ID is in FPT for the parameterization by the number of voters ($|V|$) and in XP for the parameterization by the number of voters in the solution (n').*

[2] Initially this approach is reminiscent of the (NP-hard) Longest Common Subsequence problem. However, in our problem, each candidate appears in each vote exactly once.

Proof. Suppose we are given a Hidden-ID instance (E, m', n'). We iterate through all possible size-n' subsets of voters and check if the algorithm 4 found any identity subelection consisting of at least m' candidates, if so, then we accept, otherwise we reject. The algorithm works in $O(\binom{|V|}{n'} \cdot (n' \cdot (n' \cdot |C|^2)))$ time complexity and $O(|V| \cdot |C| + |C|^2)$ space complexity. □

We show analogous results for a given set of candidates and for the parameter number of candidates.

Proposition 6. *Checking if for a given set of m' candidates there exists an identity subelection with at least n' voters is P-time solvable.*

Proof. Take a Hidden-ID instance (E, m', n') and a set of m' candidates $C' = \{c_{i_1}, c_{i_2}, \dots, c_{i_{m'}}\}$. We ask if there are n' voters that rank them identically. As there are at most $|V|$ distinct orders of candidates C' in our instance, it suffices to check if any of them appears in at least n' votes. □

The Algorithm 6 can be implemented in $O(|V| \cdot |C|)$ time and space complexity. Due to it, we obtain parameterized tractability of Hidden-ID for the number of candidates.

Corollary 7. *Hidden-ID is in FPT for the parameterization by the number of candidates ($|C|$) and in XP for the parameterization by the number of candidates in the solution (m').*

Proof. Suppose we are given a Hidden-ID instance (E, m', n'). We iterate through all size-m' sets of candidates and accept if the algorithm 6 found any identity subelection consisting of at least n' voters, otherwise we reject. The algorithm works in $O(\binom{|C|}{m'} \cdot (|V| \cdot |C|))$ time complexity and $O(|V| \cdot |C|)$ space complexity. □

The algorithms described in Propositions 4 and 6 can be used to effectively answer natural questions about preferences such as what do voters coming from a certain background mostly agree on, or are certain alternatives ranked in the same order by some large group of voters. As we showed in Corollaries 5 and 7, we are able to answer these questions effectively provided that either the number of voters or the number of candidates is relatively small (up to 20 or 30). Nevertheless, if both the number of voters and the number of candidates are not small, then this approach is too slow. To tackle this problem, we provide an ILP for Hidden-ID which finds an identity subelection (if exists) or the closest subelection to identity if such does not exist.

Proposition 8. *There is an ILP for Hidden-ID which selects a solution for a "yes"-instance and the closest subelection to identity (in terms of swap distance) for a "no"-instance.*

Proof. Let $E = (C, V)$ be the election we wish to analyze, with $C = \{c_1, \dots, c_m\}$ and $V = \{v_1, \dots, v_n\}$. All variables will be binaries. For each $i \in [n]$, we define a variable V_i with the intention that value 1 indicates that voter v_i is selected. Similarly, for each $j \in [m]$, we define a binary variable C_j with the intention that value 1 indicates that candidate c_j is selected. The variable S_{j_1,j_2} is equal to 1 if candidates C_{j_1} and C_{j_2} are selected and candidate C_{j_1} appears before C_{j_2} in the identity ranking. Variable P_{i,j_1,j_2} is equal to 1 if voter V_i agrees that C_{j_1} is ranked before C_{j_2} and both these candidates are selected. We introduce the following constraints: [3]

$$\textstyle\sum_{i\in[n]} V_i = n', \tag{1}$$

$$\textstyle\sum_{j\in[m]} C_j = m', \tag{2}$$

$$S_{j_1,j_2} + S_{j_2,j_1} = C_{j_1} \cdot C_{j_2}, \quad \forall_{j_1,j_2\in[m]}, \tag{3}$$

$$P_{i,j_1,j_2} = V_i \cdot S_{j_1,j_2}, \quad \forall_{i\in[n],j_1,j_2\in[m]}. \tag{4}$$

Constraints (1) and (2) ensure that we select the proper numbers of voters and candidates. Constraints (3) and (4) implements the logic of S and P variables, respectively. The optimization goal is to minimize: $\sum_{i\in[n],\, j_1,j_2\in[m]} P_{i,j_1,j_2} \cdot W_{i,j_1,j_2}$, where $W_{i,j_1,j_2} = [pos_{v_i}(c_{j_1}) > pos_{v_i}(c_{j_2})]$. □

We see that with the proposed ILP, we are able to maximize the solution if exists as well as to effectively determine the closest subelection if the exact one does not exist. We also consider the maximization problem for Hidden-ID, which will be useful for our experiments. For $m' \in [m]$, by Max-ID(E, m') we denote the problem of finding $n' \in [n]$ such that (n', m') is the identity signature, i.e., $size(E) = (m', n')$. We can solve it via a modification of the ILP in Proposition 8, that is, by 1) adding a constraint $\sum_{i\in[n]} V_i$ (i.e., the previous objective function) and 2) changing the optimization goal to maximize $\sum_{i\in[n]} V_i$.

3.3 Hidden Antagonism

We further analyze the problem of checking if an election contains an antagonism of a given size.

Hidden-AN:
Input: Election $E = (C, V)$, $m', n' \in \mathbb{N}$.
Question: Is there an antagonism subelection $E' = (C', V')$ of E, with $|C'| \geq m'$ and $|V'| \geq n'$?

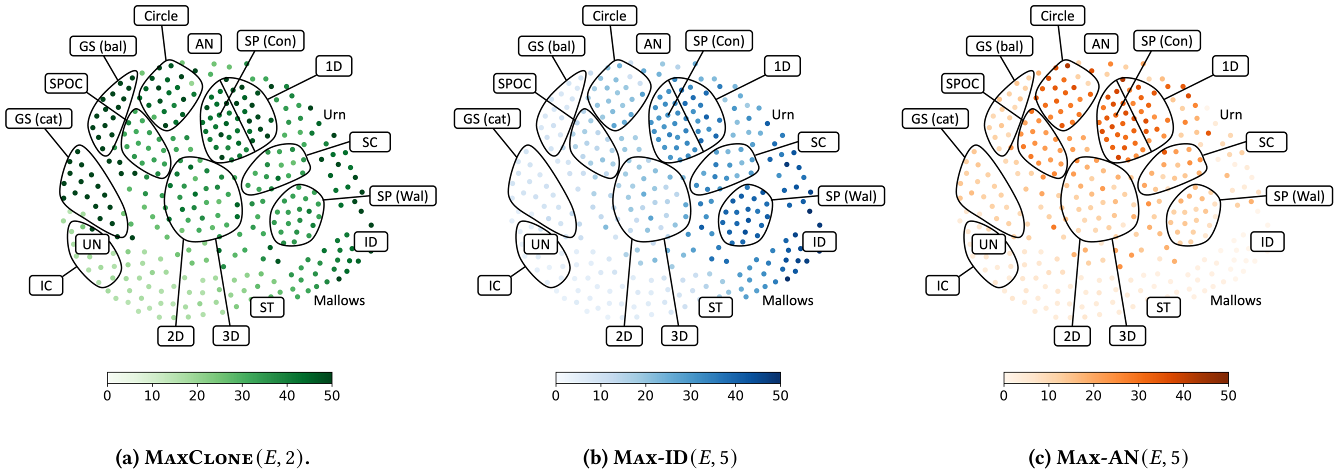
First, we show that Hidden-AN is intractable in general.

Theorem 9. *Hidden-AN is NP-complete.*

Proof Sketch. The proof extends the reduction in the proof Theorem 3. For a 3-CNF formula φ with the set of variables $X = \{x_0, \dots, x_n\}$ and the set of clauses $C = \{C_0, \dots, C_m\}$, we begin by taking the encoding provided there. Subsequently, we double the number of main voters, reversing their order of candidates corresponding to particular variables for half of these voters. So, for half of the main voters, we have that c_{x_i} and $c_{\neg x_i}$ are preferred to both c_{x_j} and $c_{\neg x_j}$, while for a half c_{x_j} and $c_{\neg x_j}$ are preferred to c_{x_i} and $c_{\neg x_i}$, if $i > j$. Observe that as in the case of the reduction in the proof of Theorem 3, by choosing a sufficient number of main voters and the required number of candidates in a target subelection as $|X'|$, we ensure that it corresponds to some valuation over X'.

Then, we take two copies of voters corresponding to each clause, requiring voters in each such copy to have a reversed order of candidates corresponding to particular variables, as in the case of main voters. Following the reasoning provided in the proof of Theorem 3, due to the reversed order of candidates, there exists an antagonism with at least $|X'|$ candidates and $2|C| + M$ voters, where M is the number of pairs of main voters, exactly when φ is satisfiable. □

[3] In some constraints, we used the multiplication operation. However, as all the variables are binaries they can be easily replaced by standard constraints of a less intuitive form.



(a) MaxClone$(E, 2)$.

(b) Max-ID$(E, 5)$

(c) Max-AN$(E, 5)$

Figure 1: Maps of elections with 10 candidates and 50 voters. Each point represents a single election, and its color represents the maximum number of voters that a) find certain two candidates clones (left), b) agree on certain five candidates being identity (middle), c) are antagonized over certain five candidates. In other words, the darker the point is, the more voters agree on a certain set of candidates being clones (left), identity (middle), or antagonism (right). On each map, ID label marks the identity election, and AN label marks the antagonism election, and dots representing elections coming from the same statistical culture were connected in clusters with names.

Although it requires more effort to model antagonized votes than identical ones, we can still compute maximum antagonism for a given set of voters or candidates in polynomial time and thus obtain FPT algorithms for Hidden-AN.

Proposition 10. *Checking if for a given set of n' voters there exists an antagonism subelection with at least m' candidates is P-time solvable.*

Proof. Suppose we are given a Hidden-AN instance (E, m', n') and a set of n' voters $V' = \{v_{i_1}, v_{i_2}, \ldots, v_{i_{n'}}\}$. We ask if there exists a set of m' candidates such that all voters from V' are antagonized over them. That is, half of them rank these candidates in the same order, whereas the other half in the opposite one.

For $m' = 1$ the answer is naturally yes, so we assume that $m' \geq 2$. Suppose now that the answer is yes, the solution is order p' of candidates $C' \subseteq C$, candidate c_b is at the beginning of p', and candidate c_e is at the end of p'. Let $V'_{c_b \succ c_e}$ and $V'_{c_e \succ c_b}$ be the voters from V' preferring c_b over c_e and c_e over c_b, respectively. Then all voters from $V'_{c_b \succ c_e}$ must order candidates C' in the order p' and all voters from $V'_{c_e \succ c_b}$ must order them exactly in the reversed order r'. However, all voters from $V'_{c_e \succ c_b}$ order candidates C' exactly in the reversed order r' if and only if they order them in the order p' after reversing their whole votes.

With this observation, we propose a polynomial-time algorithm as follows. We iterate over each pair of distinct candidates c_b, c_e from C (with the intention that they will be, respectively, the first and the last candidate in the antagonism subelection). If $|V'_{c_b \succ c_e}| \neq |V'_{c_e \succ c_b}|$, then the sets of antagonized voters do not have the same cardinality, so we continue with the next pair. Now we know that $|V'_{c_b \succ c_e}| = |V'_{c_e \succ c_b}|$. We remove all candidates from the votes that appear in at least one vote not between c_b and c_e (i.e., we keep candidate c if and only if for each vote either $c_b \succ c \succ c_e$ or $c_e \succ c \succ c_b$). We are now left with the truncated votes consisting of c_b and c_e ranked at the extreme positions and all remaining candidates ranked between them in all votes. If we end up with less than m' candidates, then we continue with the next pair. We reverse votes in which $c_e \succ c_b$, use the algorithm 4 to see if there is an identity subelection with m' candidates, and accept if there is one (note that due to $|V'_{c_b \succ c_e}| = |V'_{c_e \succ c_b}|$ it is equivalent to having half of voters approving one order and the other half preferring the opposite one). We reject if we do not find any solution for any pair of candidates $c_b, c_e \in C$. The algorithm runs in $O(n' \cdot |C| + |C|^2 \cdot (n' \cdot |C|^2))$ time and uses $O(n' \cdot |C| + |C|^2)$ space. □

Proposition 11. *Checking if for a given set of m' candidates there exists an antagonism subelection with at least n' voters is P-time solvable.*

Proof. Suppose we are given a Hidden-AN instance (E, m', n') and a set of m' candidates $C' = \{c_{i_1}, c_{i_2}, \ldots, c_{i_{m'}}\}$. We ask if there exists a set of n' voters that are antagonized over them, that is, half of them rank candidates C' in one way and the other half in the opposite order. We create a hash map D mapping orders (permutations) of candidates from C' to lists of voters that rank them in this order. For the sake of brevity, let $D[p]$ be the value in D associated with the key p, i.e., the list of voters that rank candidates C' in the order p. Then, we iterate over each order p, in the votes of D and accept if for any order p, both $D[p]$ and $D[r]$ contain at least $n'/2$ voters where r is the reversed order of p. Analogously to the algorithm in Proposition 6, the algorithm runs in polynomial time because we consider only permutations that appear in the given votes. Specifically, its time and space complexity is $O(|V| \cdot |C|)$. □

Then, fixed-parameter tractability of Hidden-AN for the number of voters and the number of candidates follows.

Corollary 12. *Hidden-AN is in FPT parametrized by the number of voters ($|V|$) or candidates ($|C|$) as well as in XP for the parameterization by the number of voters (n') or candidates (m') in the solution.*

Analogously to Hidden-ID, these FPT algorithms suffice if either the number of voters or the number of candidates is small, but they are too small if both of these values are not small. To handle it, we use a simple ILP for that problem. (The details are in the full version of the paper).

Proposition 13. *There is an ILP for Hidden-AN which selects a solution for a "yes"-instance and the closest subelection to antagonism (in terms of swap distance) for a "no"-instance.*

The strong advantage of this ILP is that it also manages situations in which the desired antagonism does not exist, which, as we will see in experiments, is not a rare case.

Additionally, we study a maximization version of Hidden-AN that will be crucial regarding our experiments. Given $m' \in [m]$, by Max-AN(E, m') we denote the problem of finding $n' \in [n]$ such that (n', m') is the antagonism signature, i.e., $size(E) = (m', n')$.

We note that the ILPs provided in this section, i.e., in Propositions 8 and 13, as well as those for Max-ID and Max-AN, are crucial for the experiments we provide in the next section.

4 EXPERIMENTS

In this section, we focus on the practical application of our approach. First, we study our problems with synthetic data. Later on, we analyze several real-life instances.

4.1 Map of Elections

To depict our experimental results, we use the framework introduced by Szufa et al. [18] and extended by Boehmer et al. [3], known as *map of elections*. The map serves us to better understand the space of elections and is particularly useful when conducting experiments. Each point on the map depicts a single election. The embeddings were calculated based on the mutual distances between elections (computed with some distance function). The closer the two points on the map are, the more similar the elections these points depict. For instance, elections coming from the same statistical cultures or similar models are often clones on the map, while elections coming from very different models are usually more distant on the map. In this case, we use the map from Boehmer et al. [4] that consists of 344 elections[4] with 10 candidates and 50 voters. The map is based on the isomorphic swap distance [11], and is embedded using Fruchterman-Reingold algorithm [13].

Having prepared the map and its points, one can put on them some properties hidden in the colors or shapes of points. E.g., Szufa et al. [17] color the points on the map of approval elections with several statistics (e.g., cohesiveness level, PAV run time, and maximum approval score) to see how well statistical cultures and elections situated in different regions satisfy these properties or how well they perform. Here we will conduct a similar analysis of values distribution as well as its location on the map. We also investigate why this particular coloring occurred and what it means.

For each election from the given dataset, we computed the three following characteristics: (1) Max-ID$(E, 5)$, (2) Max-AN$(E, 5)$, (3) MaxClone$(E, 2)$. The results are presented in Figure 1. (Complementary results, containing separate average values for each statistical culture, are presented in the full version of the paper).

In the case of MaxClone$(E, 2)$ (as depicted in the leftmost map) the darker the point, the more voters agree that there exists a pair of clone candidates. The most modest values are consistently seen in elections originating from the impartial culture (i.e., each vote is sampled uniformly at random), averaging at a value of 17, with a remarkably low standard deviation of only 1.22. Note that the value for MaxClone$(E, 2)$ seems not to be strongly correlated with the position on the map. It is because the map is based on swap distance. By making relatively few swaps (i.e., increasing a distance just a bit), we can significantly decrease the number of voters agreeing that two candidates are clones. In other words, given an election E we can create a new election E' that is very close to E (distance-wise) but has a much lower MaxClone value. In principle, for the MaxClone problem, usually by one swap we can lower the results by one (unless that swap is creating a new solution involving a different set of candidates).

For the Max-ID$(E, 5)$ (the middle map) we observe a strong correlation between the number of voters agreeing on given five candidates being ranked in a particular order and the swap distance from ID (the Pearson correlation coefficient (PCC) is -0.791). Similarly, the results for Max-AN$(E, 5)$ (the rightmost map) are strongly correlated with the distance from AN (PCC = -0.845). Both of these correlations are reasonable, as the larger the hidden identity (resp. antagonism), the fewer swaps we need to convert the election into ID (resp. AN). This means that for Max-ID and Max-ID there is a strong correlation between the position on the map and the size of the signature. Unlike for clones, for identity and antagonism, it is harder to "spoil" the inner substructure, especially when m' is relatively small with regard to m. For instance, for a vote $a \succ b \succ c \succ d \succ e \succ f$, no matter which three candidates are forming the solution, we can always "spoil" this vote with just one swap. And for example for identity if a, c, and f form a solution, we need at least two swaps to "spoil" this vote.

Remark 1. *The way we define Max-AN might seem too rigid, as we require exactly the same number of "base" and "reverse" votes. However, we have also verified two other approaches. One, where we maximized the sum of #base and #reverse votes; it turns out, that usually the outcome is more similar to the one provided by Max-ID than by Max-AN. The second approach is to use the product of #base and #reverse votes. There, the result was usually very similar to the one provided by the "rigid" approach (PCC = 0.929), yet the running-time was significantly longer. Therefore, for the sake of simplicity, we focus on the "rigid" approach.*

4.2 Real-Life Instances

We conducted experiments on real-life instances. In particular, we focus on two datasets, i.e., the *Sushi* dataset, where 5000 people expressed their preferences over 10 sushi types [16]; and the *Grenoble* dataset, containing data from a field experiment held in Grenoble in 2017, where people expressed their preferences over French presidential candidates [5]. There, we use the same data preprocessing method as Faliszewski et al. [10].

[4] Detailed description is provided in the full version of the paper.

We run MaxClone, Max-ID, and Max-AN for all possible numbers of hidden candidates. The results are presented in Figure 2. For all experiments, we include the respective values from impartial culture (IC) instances, which we treat as a lower bound.[5](Each value is an average derived from 10 different IC elections).

We first focus on hidden clones. Every individual candidate is seen as a clone by all the voters. However, it is worth noting that if the number of hidden candidates is equal to the number of all candidates, then again all the voters would agree that all the candidates are clones, hence the characteristic "U" shape in Figure 2.

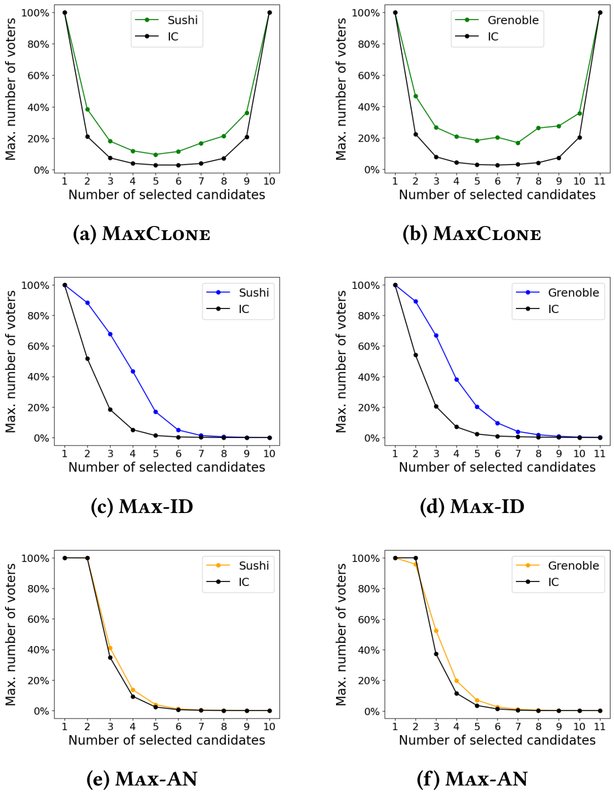
For Sushi, 38.5% of voters agree that Tamago (egg) and Kappa-maki (cucumber roll) are clones. This is especially interesting, as these are the only two vegetarian options in that dataset. For Grenoble, the strongest set of two clones is Nathalie Arthaud and Philippe Poutou (clones for 46.7% of voters), which can be explained by the fact that they have similar far-left ideologies; and the strongest clone set of size 3 is composed of Jean-Luc Mélenchon (left), Benoît Hamon (centre-left) and Emmanuel Macron (centre-right), who are clones for 26.6% of the voters. This may initially look surprising but can be explained knowing that (a) all three candidates are major, well-known candidates; (b) on the left-right axis, they are arguably contiguous; (c) there was (at least in the voters of the dataset) a clear dividing line between the candidates from left to center-right on the one hand, and right and far-right candidates on the other.

We now shift to hidden identity. In the case of Sushi, an overwhelming 88.3% of the voters agree that Toro (fatty tuna) is better than Kappa-maki (cucumber roll). What is more intriguing is that 43.4% of the voters agree on the following ranking: Toro (fatty tuna) is preferred over Maguro (tuna), which is, in turn, preferred over Tekka-maki (tuna roll), and finally, Kappa-maki (cucumber roll). This order of preference is especially interesting as it mirrors the price hierarchy of these sushi types. When it comes to the political elections dataset, 89.2% of the voters prefer Benoît Hamon (center-left) to Marine Le Pen (far-right).

As to the hidden antagonism, we only briefly discuss the results for the sushi dataset. The pair Ika (squid) and Tekka-maki (tuna roll) antagonized the whole society (99.9% of voters). Moreover, 41.2% of the voters agree or strongly disagree (casting reverse order) with the following ranking: Uni (sea urchin) ≻ Kappa-maki (cucumber roll) ≻ Tamago (egg). It is intriguing that both the Sushi and Grenoble datasets show minimal signs of antagonism. The results are almost the same as for IC elections.

5 SUMMARY AND FUTURE WORK

We explored the concept of hidden substructures in ordinal elections. We focused on three types of consistency: Identity, antagonism, and clones. We executed a comprehensive analysis of the complexity of the introduced problems and provided algorithms that can be used in practice. We showed as a possible direction the search for the closest subelection to the desired one if the exact one does not exist. Furthermore, we provided experimental evaluations on synthetic and real-life datasets. The experiments on real-life datasets confirmed that identifying consistent subelections indeed helps in learning interesting information hidden in an election.



(a) MaxClone

(b) MaxClone

(c) Max-ID

(d) Max-ID

(e) Max-AN

(f) Max-AN

Figure 2: Comparison of Sushi and Grenoble datasets. The black lines denote the results for impartial culture elections.

Analyzing substructures of elections can help better understand different segments of the population and their preferences, which can be beneficial in a variety of contexts, such as consumer behavior or political opinion analysis.

We see this paper as a starting point for a more thorough study. Indeed, there are numerous possibilities for further research related to the problems we considered. To start with, we could soften our criteria and look for subelections that are near-identity or near-antagonism, and look for near-clones. Next, the complexity of these issues could be analyzed when applied to structured domains, such as single-peaked or single-crossing domains. A further natural extension could involve exploring approval elections.

ACKNOWLEDGEMENTS

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), and from the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). We thank Florian Yger for fruitful discussions.

[5] Technically, it is possible to create an instance with even smaller clones/identity/antagonism than impartial culture, but the difference will not be of great importance.

REFERENCES

[1] J. Alcalde-Unzu and M. Vorsatz. 2013. Measuring the cohesiveness of preferences: an axiomatic analysis. *Soc. Choice Welf.* 41, 4 (2013), 965–988.

[2] J. C. Rodriguez Alcantud, R. de Andrés Calle, and J. Manuel Cascón. 2013. On measures of cohesiveness under dichotomous opinions: Some characterizations of approval consensus measures. *Inf. Sci.* 240 (2013), 45–55.

[3] N. Boehmer, R. Bredereck, P. Faliszewski, R. Niedermeier, and S. Szufa. 2021. Putting a Compass on the Map of Elections. In *Proceedings of IJCAI-2021*. 59–65.

[4] N. Boehmer, P. Faliszewski, R. Niedermeier, S. Szufa, and T. Was. 2022. Understanding Distance Measures Among Elections. In *Proceedings of IJCAI-2022*. 102–108.

[5] S. Bouveret, R. Blanch, A. Baujard, F. Durand, H. Igersheim, J. Lang, A. Laruelle, J.-F. Laslier, I. Lebon, and V. Merlin. 2018. Voter Autrement 2017 - Online Experiment. Dataset and companion article on Zenodo.

[6] R. Colley, U. Grandi, C. A. Hidalgo, M. Macedo, and C. Navarrete. 2023. Measuring and Controlling Divisiveness in Rank Aggregation. In *Proceedings of IJCAI-2023*.

[7] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. 2022. *Introduction to Algorithms* (fourth ed.). MIT Press/McGraw Hill.

[8] E. Elkind, P. Faliszewski, and A. M. Slinko. 2012. Clone structures in voters' preferences. In *EC*. ACM, 496–513.

[9] E. Elkind, M. Lackner, and D. Peters. 2022. Preference Restrictions in Computational Social Choice: A Survey. *CoRR* abs/2205.09092 (2022).

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

[11] P. Faliszewski, P. Skowron, A. Slinko, S. Szufa, and N. Talmon. 2019. How Similar Are Two Elections?. In *Proceedings of AAAI-2019*. 1909–1916.

[12] P. Faliszewski, K. Sornat, and S. Szufa. 2022. The Complexity of Subelection Isomorphism Problems. In *Proceedings of AAAI-2022*. 4991–4998.

[13] T. Fruchterman and E. Reingold. 1991. Graph drawing by force-directed placement. *Software: Practice and Experience* 21, 11 (1991), 1129–1164.

[14] G. Govaert and M. Nadif. 2014. *Co-Clustering*. ISTE-Wiley. http://hal.archives-ouvertes.fr/hal-00933301

[15] V. Hashemi and U. Endriss. 2014. Measuring Diversity of Preferences in a Group. In *ECAI (Frontiers in Artificial Intelligence and Applications, Vol. 263)*. IOS Press, 423–428.

[16] T. Kamishima. 2003. Nantonac Collaborative Filtering: Recommendation Based on Order Responses. In *Proceedings of KDD-2003*. 583–588.

[17] S. Szufa, P. Faliszewski, L. Janeczko, M. Lackner, A. Slinko, K. Sornat, and N. Talmon. 2022. How to Sample Approval Elections?. In *Proceedings of IJCAI-2022*. 496–502.

[18] S. Szufa, P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. 2020. Drawing a Map of Elections in the Space of Statistical Cultures. In *Proceedings of AAMAS-20*. 1341–1349.