The Complexity of Subelection Isomorphism Problems

Piotr Faliszewski,[1] Krzysztof Sornat,[1] Stanisław Szufa[1,2]

[1]AGH University, Kraków, Poland
[2]Jagiellonian University, Kraków, Poland
{faliszew, sornat, szufa}@agh.edu.pl

Abstract

We study extensions of the Election Isomorphism problem, focused on the existence of isomorphic subelections. Specifically, we propose the SUBELECTION ISOMORPHISM and the MAXIMUM COMMON SUBELECTION problems and study their computational complexity and approximability. Using our problems in experiments, we provide some insights into the nature of several statistical models of elections.

1 Introduction

We study the computational complexity of several extensions of the ELECTION ISOMORPHISM problem, recently introduced by Faliszewski et al. (2019) as an analogue of GRAPH ISOMORPHISM. While in the latter we are given two graphs and we ask if they can be made identical by renaming the vertices, in the former we are given two ordinal elections (i.e., elections where each voter ranks the candidates from the most to the least appealing one) and ask if they can be made identical by renaming the candidates and reordering the voters. Interestingly, even though the exact complexity of GRAPH ISOMORPHISM, as well as of many related problems, remains elusive (Babai et al. 2015), ELECTION ISOMORPHISM has a simple polynomial-time algorithm (Faliszewski et al. 2019). Yet, in many practical settings perfect isomorphism is too stringent and approximate variants are necessary. For the case of GRAPH ISOMORPHISM, researchers considered two types of relaxations: Either they focused on making a small number of modifications to the input graphs that make them isomorphic (see, e.g., the works of Arvind et al. (2012) and Grohe, Rattan, and Woeginger (2018)), or they sought (maximum) isomorphic subgraphs of the input ones (see, e.g., the classic paper of Cook (1971) and the textbook of Garey and Johnson (1979); for an overview focused on applications in cheminformatics we point to the work of Raymond and Willett (2002)). For the case of elections, the former approach was already taken by Faliszewski et al. (2019) and several other researchers (Vayer et al. 2020; Szufa et al. 2020; Boehmer et al. 2021). Our goal is to explore the latter route.

More specifically, we consider the SUBELECTION ISOMORPHISM and MAXIMUM COMMON SUBELECTION families of problems. In the former, we are given two elections, a smaller and a larger one, and we ask if it is possible to delete some candidates and voters from the larger election so that it becomes isomorphic to the smaller one. Put differently, we ask if the smaller election occurs as a minor in the larger one. One reason why this problem is interesting is its connection to restricted preference domains. For example, both single-peaked (Black 1958) and single-crossing (Mirrlees 1971; Roberts 1977) elections are characterized as those that do not have certain forbidden minors (Ballester and Haeringer 2011; Bredereck, Chen, and Woeginger 2013). We show that SUBELECTION ISOMORPHISM is NP-complete and W[1]-hard for the parameterization by the size of the smaller election, which suggests that there are no fast algorithms for the problem. Fortunately, the characterizations of single-peaked and single-crossing elections use minors of constant size and, such elections can be recognized efficiently; indeed, there are very fast algorithms for these tasks (Bartholdi and Trick 1986; Escoffier, Lang, and Öztürk 2008; Elkind, Faliszewski, and Slinko 2012). Our results show that characterizations with non-constant minors might lead to NP-hard recognition problems.

In our second problem, MAXIMUM COMMON SUBELECTION, we ask for the largest isomorphic subelections of the two input ones. Since their size can be used as a (particularly demanding) measure of similarity, our work is related to those of Faliszewski et al. (2019) and Szufa et al. (2020). In the former, the authors define the similarity between elections using variants of the swap distance and the Spearman footrule (and find these measures to be intractable), whereas in the latter the authors propose a polynomial-time computable measure, based on analyzing the frequency with which candidates appear on particular positions in the votes. While we find that many of our problems are NP-hard (and hard to approximate), we also find polynomial algorithms, also for practically useful cases.

For both our problems, we consider their "candidate" and "voter" variants. For example, in CANDIDATE SUBELECTION ISOMORPHISM we ask if it is possible to delete candidates from the larger election (but without deleting any voters) so that it becomes isomorphic with the smaller one. Similarly, in MAXIMUM COMMON VOTER-SUBELECTION we ask if we can ensure isomorphism of the two input elections by only deleting voters (so that at least a given number of voters remains). In Section 5 we use this latter problem to evaluate similarity between elections generated from various statistical cultures. These results confirm some findings observed by Szufa et al. (2020) and Boehmer et al. (2021) in their "maps of elections" and give a new perspective on some of these statistical cultures.

In the most general variants of our problems, we assume that both input elections are over different candidate sets and include different voters. Yet, sometimes it is natural to assume that the candidates or the voters are the same (for example, in a presidential election votes collected in two different districts would have the same candidate sets, but different voters, whereas two consecutive presidential elections would largely involve the same voters, but not necessarily the same candidates). We model such scenarios by variants of our problems where either the matchings between the candidates or the voters of the input elections are given (indeed, this way we follow the approach of Faliszewski et al. (2019)). While one would expect that having such matchings would make our problems easier, there are cases where they remain NP-hard even with both matchings. This stands in sharp contrast to the results of Faliszewski et al. (2019). For a summary of our results, see Table 1.

2 Preliminaries

For a positive integer k, we write $[k]$ to denote the set $\{1, \dots, k\}$, and by S_k we refer to the set of all permutation over $[k]$. Given a graph G, we write $V(G)$ to refer to its set of vertices and $E(G)$ to refer to its set of edges.

Elections. An election is a pair $E = (C, V)$ that consists of a set $C = \{c_1, \dots, c_m\}$ of candidates and a collection $V = (v_1, \dots, v_n)$ of voters. Each voter $v \in V$ has a preference order, i.e., a ranking of the candidates from the most to the least appreciated one, denoted as $\succ_v$. Given two candidates $c_i, c_j \in C$, we write $c_i \succ_v c_j$ (or, equivalently, $v\colon c_i \succ c_j$) to denote that v prefers c_i to c_j. We extend this notation to more than two candidates in a natural way. For example, we write $v\colon c_1 \succ c_2 \succ \cdots \succ c_m$ to indicate that voter v likes c_1 best, then c_2, and so on, until c_m. If we put some set S of candidates in such a description of a preference order, then we mean listing its members in some arbitrary (but fixed, global) order. Including $\overleftarrow{S}$ means listing the members of S in the reverse of this order. We often refer to the preference orders as either the votes or the voters, but the exact meaning will always be clear from the context. By the *size* of an election, we mean the number of candidates multiplied by the number of voters. Occasionally we discuss single-peaked elections.

Definition 2.1 (Black (1958))**.** *Let C be a set of candidates, and let $\lhd$ be a linear order over C (referred to as the societal axis). We say that a vote v is single-peaked with respect to $\lhd$ if for each $k \in [m]$, the top k candidates in v form an interval within $\lhd$. An election is single-peaked if there is a societal axis for which all its votes are single-peaked.*

Given elections $E = (C, V)$ and $E' = (C', V')$, we say that E' is a *subelection* of E if C' is a subset of C and V' can be obtained from V by deleting some voters and restricting the remaining ones to the candidates from C'. We say that E' is a *voter subelection* of E if we can obtain it by only deleting voters from E, and that E' is a *candidate subelection* of E if we can obtain it from E by only deleting candidates.

Election Isomorphism. Let E be an election with candidate set $C = \{c_1, \dots, c_m\}$ and voter collection $V = (v_1, \dots, v_n)$. Further, let D be another set of m candidates and let $\sigma\colon C \to D$ be a bijection between C and D. For each voter $v \in V$, by $\sigma(v)$ we mean a voter with the same preference order as v, except that each candidate c is replaced with $\sigma(c)$. By $\sigma(V)$ we mean voter collection $(\sigma(v_1), \dots, \sigma(v_n))$. Similarly, given a permutation $\pi \in S_n$, by $\pi(V)$ we mean $(v_{\pi(1)}, \dots, v_{\pi(n)})$.

Two elections are *isomorphic* if it is possible to rename their candidates and reorder their voters so that they become identical (Faliszewski et al. 2019). Formally, elections (C_1, V_1) and (C_2, V_2), are isomorphic if $|C_1| = |C_2|$, $|V_1| = |V_2|$, and there is a bijection $\sigma\colon C_1 \to C_2$ and a permutation $\pi \in S_{|V_1|}$ such that $(\sigma(C_1), \sigma(\pi(V_1))) = (C_2, V_2)$. We refer to σ as the candidate matching and to π as the voter matching. In the Election Isomorphism problem we are given two elections and we ask if they are isomorphic.

Computational Complexity. We assume familiarity with (parameterized) computational complexity theory; for background, we point the readers to the textbooks of Papadimitriou (1994) and Cygan et al. (2015). Most of our intractability proofs follow by reductions from the Clique problem. An instance of Clique consists of a graph G and a nonnegative integer k, and we ask if G contains k vertices that are all connected to each other. Clique is well-known to be both NP-complete and W[1]-complete, for the parameterization by k (Downey and Fellows 1995). Additionally, we provide some lower-bounds based on the Exponential Time Hypothesis (ETH), which is a popular conjecture on solving the CNF-SAT problem. For a formal statement see, e.g., Conjecture 14.1 in (Cygan et al. 2015). Specifically, in our lower-bound proofs we use a consequence of the ETH which says that there is no $|V(G)|^{o(k)}$-time algorithm for Clique (Chen et al. 2006). As all the problems that we study can easily be seen to belong to NP, in our NP-completeness proofs we only give hardness arguments.

3 Variants of the Isomorphism Problem

We introduce two extensions of the Election Isomorphism problem, inspired by Subgraph Isomorphism and Maximum Common Subgraph. In the former, we are given two elections and we ask if the smaller one is isomorphic to a subelection of the larger one. That is, we ask if we can remove some candidates and voters from the larger election to make the two elections isomorphic.

Definition 3.1. *An instance of* Subelection Isomorphism *consists of two elections, $E_1 = (C_1, V_1)$ and $E_2 = (C_2, V_2)$, such that $|C_1| \leq |C_2|$ and $|V_1| \leq |V_2|$. We ask if there is a subelection E' of E_2 isomorphic to E_1.*

The Voter-Subelection Isomorphism problem is defined in the same way, except that we require E' to

Problem	no matching	voter-matching	cand.-matching	both matchings
Election Isomorphism	P	P	P	P
Subelection Isomorphism	W[1]-h. [Thm. 3]	NP-com. [Thm. 5]	P [Thm. 2]	P [Thm. 2]
Cand.-Subelection Isomorphism	NP-com. [Prop. 6]	NP-com. [Thm. 5]	P [Thm. 2]	P [Thm. 2]
Voter-Subelection Isomorphism	P [Thm. 2]	P [Thm. 2]	P [Thm. 2]	P [Thm. 2]
Max. Common Subelection	W[1]-h. [Prop. 11]	NP-com. [Prop. 11]	NP-com. [Prop. 11]	NP-com. [Prop. 11]
Max. Common Cand.-Subelection	NP-com. [Prop. 10]	NP-com. [Prop. 10]	NP-com. [Prop. 10]	W[1]-com. [Thm. 7]
Max. Common Voter-Subelection	P [Thm. 2]	P [Thm. 2]	P [Thm. 2]	P [Thm. 2]

Table 1: An overview of our results; those for Election Isomorphism are due to Faliszewski et al. (2019). W[1]-hardness holds with respect to the size of the smaller election or a common subelection. Indicated W[1]-hard problems are also NP-hard.

be a voter subelection of E_2. Similarly, in Candidate-Subelection Isomorphism we require E' to be a candidate subelection. We often abbreviate the name of the latter problem to Cand.-Subelection Isomorphism.

Example 3.1. *Consider elections $E = (C, V)$ and $F = (D, U)$, where $C = \{a, b, c\}$, $D = \{x, y, z, w\}$, $V = (v_1, v_2, v_3)$ and $U = (u_1, u_2, u_3)$, with preference orders:*

$$\begin{array}{ll} v_1\colon a \succ b \succ c, & u_1\colon w \succ x \succ y \succ z, \\ v_2\colon b \succ a \succ c, & u_2\colon y \succ w \succ x \succ z, \\ v_3\colon c \succ b \succ a, \text{ and} & u_3\colon z \succ w \succ y \succ x. \end{array}$$

If we remove candidate w from (D, U), then we find that the resulting elections are isomorphic (to see this, it suffices to match voters v_1, v_2, v_3 with u_1, u_2, u_3, respectively, and candidates a, b, c with x, y, z). Thus E is isomorphic to a (candidate) subelection of F and, so, (E, F) is a yes-*instance of* (Cand.-)Subelection Isomorphism.

In the Maximum Common Subelection problem we seek the largest isomorphic subelections of two given ones. We often abbreviate Maximum as Max.

Definition 3.2. *An instance of* Max. Common Subelection *consists of two elections, $E_1 = (C_1, V_1)$ and $E_2 = (C_2, V_2)$, and a positive integer t. We ask if there is a subelection E'_1 of E_1 and a subelection E'_2 of E_2 such that E'_1 and E'_2 are isomorphic and the size of E'_1 (or, equivalently, the size of E'_2) is at least t.*

Analogously to the case of Subelection Isomorphism, we also consider the Max. Common Cand.-Subelection and Max. Common Voter-Subelection problems. In the former, E'_1 and E'_2 must be candidate subelections and in the latter they need to be voter subelections (thus in the former problem E_1 and E_2 must have the same numbers of voters, and in the latter E_1 and E_2 must have the same numbers of candidates).

For each of the above-defined problems we consider its variant with or without the candidate or voter matching. Specifically, the variants defined above are *with no matchings*. Variants *with candidate matching* include a bijection σ that matches (some of) the candidates in one election to (some of) those in the other (in case of Subelection Isomorphism and its variants, all the candidates from the smaller election must be matched to those in the larger one; in case of Max. Common Subelection there are no such requirements). Then we ask for an isomorphism between respective subelections that agrees with σ. In particular, this means that none of the unmatched candidates remain in the considered subelections (another interpretation is to assume that both input election have the same candidate sets).

Example 3.2. *Consider elections (C, V) and (D, U) from Example 3.1, and a matching σ such that $\sigma(a) = x, \sigma(b) = w$, where c, y, and z are unmatched. After applying it and dropping the unmatched candidates, the votes in the first election become $v_1\colon x \succ w$, $v_2\colon w \succ x$, and $v_3\colon w \succ x$, whereas all the voters in the second election have preference order $w \succ x$. Thus, this instance of* Max. Common Subelection with Candidate Matching *has isomorphic subelections, respecting the matching σ, of size $2 \times 2 = 4$.*

Variants *with voter matching* are defined similarly: We are given a matching between (some of) the voters from one election and (some of) the voters from the other (and, again, for Subelection Isomorphism and its variants, each voter from the smaller election is matched to some voter in the larger one). The sought-after isomorphism must respect this matching (again, this means that we can disregard the unmatched voters).

Variants *with both matchings* include both the matching between the candidates and the matching between the voters (note that these variants are not trivial because we still need to decide who to delete). By writing *all four matching cases* we mean the four just-described variants of a given problem.

Finally, we note that each variant of Max. Common Subelection is at least as computationally difficult as its corresponding variant of Subelection Isomorphism.

Proposition 1. *Let M be a variant of* Max. Common Subelection *and let S be a corresponding variant of* Subelection Isomorphism. *We have that S many-one reduces to M in polynomial time.*

4 Computational Complexity Analysis

In this section, we present our complexity results. While in most cases we obtain intractability (see Table 1 for a summary of our results), we find that all our problems focused on voter subelections are solvable in polynomial time, and having candidate matchings leads to polynomial-time algorithm for all the variants of Subelection Isomorphism. Missing proofs are available in the full version of the paper (Faliszewski, Sornat, and Szufa 2021).

All our polynomial-time results rely on the trick used by Faliszewski et al. (2019) for the case of ELECTION ISOMORPHISM. The idea is to guess a pair of (matched) voters and use them to derive the candidate matching.

Theorem 2. VOTER-SUBELECTION ISOMORPHISM *and* MAX. COMMON VOTER-SUBELECTION *are in* P *for all four matching cases.* SUBELECTION ISOMORPHISM, CAND.-SUBELECTION ISOMORPHISM *are in* P *for cases with candidate matchings.*

4.1 Intractability of Subelection Isomorphism

Next we show computational hardness for all the remaining variants of our problems. In this section we consider SUBELECTION ISOMORPHISM.

Theorem 3. SUBELECTION ISOMORPHISM *is* NP*-complete and* W[1]*-hard with respect to the size of the smaller election.*

Proof. Before we describe our reduction, we first provide a method for transforming a graph into an election: For a graph H, we let E_H be an election whose candidate set consists of the vertices of H and two special candidates, α_H and β_H, and whose voters correspond to the edges of H. Specifically, for each edge $e = \{x, y\} \in E(H)$ we have four voters, $v_e^1, \dots v_e^4$, with preference orders:

$$v_e^1 \colon x \succ y \succ \alpha_H \succ \beta_H \succ V(H) \setminus \{x, y\},$$
$$v_e^2 \colon x \succ y \succ \beta_H \succ \alpha_H \succ V(H) \setminus \{x, y\},$$
$$v_e^3 \colon y \succ x \succ \alpha_H \succ \beta_H \succ V(H) \setminus \{x, y\},$$
$$v_e^4 \colon y \succ x \succ \beta_H \succ \alpha_H \succ V(H) \setminus \{x, y\}.$$

We give a reduction from CLIQUE. Given an instance (G, k) of CLIQUE, where G has at least k vertices and $\binom{k}{2}$ edges, we let K be a size-k complete graph and we form an instance (E_K, E_G) of SUBELECTION ISOMORPHISM. The reduction runs in polynomial time and it remains to show correctness.

First, let us assume that G has a size-k clique. Let X be the set of its vertices and let Y be the set of its edges. We form a subelection E' of E_G by removing all the candidates that are not in $X \cup \{\alpha_G, \beta_G\}$ and removing all the voters that do not correspond to the edges from Y. One can verify that E' and E_K are, indeed, isomorphic.

Second, let us assume that E_K is isomorphic to some subelection E' of E_G. We will show that this implies that G has a size-k clique. First, we claim that E' includes both α_G and β_G. To see why this is so, consider the following two cases:

1. If E' contained exactly one of α_G, β_G, then this candidate would appear in every vote in E' among the top three positions. Yet, in E_K there is no candidate with this property so E' and E_K would not be isomorphic.
2. If E' contained neither α_G nor β_G then every vote in E' would rank some vertex candidates z and w on positions three and four (to be able to match α_K and β_K to them). However, by the construction of E_G, either in every vote from E' we would have $z \succ w$ or in every vote from E' we would have $w \succ z$. Since in E_K half of the voters rank the candidates from positions three and four in the opposite way, E' and E_K would not be isomorphic.

Thus α_G and β_G are included in E'. Moreover α_G and β_G are matched with α_K and β_K because they are the only candidates from E_G that can appear on positions three and four in every vote in E' but possibly in different order. As a consequence, for each vote v from E_G that appears in E', the candidate set of E' must include the two candidates from $V(G)$ that v ranks on top (if it were not the case, then E' would contain a candidate—either α_G or β_G—that appeared in all the votes within the top four positions and in some vote within top two positions; yet, E_K does not have such a candidate). This means that for each edge $e \in E(G)$, if E' contains some voter v_e^i for $i \in [4]$, then it also contains the other voters corresponding to e (otherwise E' and E_K would not be isomorphic). Because the number of voters in E_K is $4\binom{k}{2}$, hence the number of distinct chosen (in E') corresponding edges from G equals $\binom{k}{2}$. As said before, for each such chosen edge, we also choose two corresponding vertices as candidates. It means that the number of chosen candidates (except α_G and β_G) is between k and $2\binom{k}{2}$. However, the number of candidates in E_K except α_K and β_K is k, therefore we conclude that chosen vertex-candidates form a size-k clique in G. This completes the proof of NP-hardness.

To show W[1]-hardness, note that the number of candidates and voters in the smaller election equals $k+2$ and $4\binom{k}{2}$ respectively, hence the size of the smaller election is a function of parameter k for which CLIQUE is W[1]-hard. □

The above reduction shows something stronger than the theorem claims. Indeed, assuming ETH one cannot compute a solution faster than a straightforward brute-force approach.

Proposition 4. SUBELECTION ISOMORPHISM *has an* $O^*(m^k)$*-time algorithm, where* k *is the number of candidates in the smaller election and* m *is the number of candidates in the larger election (hence* $\mathrm{XP}(k)$*). Moreover, assuming ETH, there is no* $O^*(m^{o(k)})$*-time algorithm.*

As a consequence of the above XP algorithm, even testing if a fairly small, constant-sized, election is isomorphic to a subelection of some bigger one may require a polynomial-time algorithm with impractically large exponents. Luckily, in practice this is not always the case. For example, single-peaked elections are characterized as exactly those that do not have subelections isomorphic to certain two elections of sizes 8 and 9 (Ballester and Haeringer 2011), but there is an algorithm for deciding if a given election is single-peaked that runs in time $O(nm)$, where m is the number of candidates and n is the number of voters (Escoffier, Lang, and Öztürk 2008). For single-crossing elections, such a characterization uses subelections of sizes up to 18, but there is a recognition algorithm that runs in time $O(n^2 + nm \log m)$ (Doignon and Falmagne 1994), which can even be improved using appropriate data structures.

Next we consider CAND.-SUBELECTION ISOMORPHISM. In this problem both elections have the same number of voters and we ask if we can delete candidates from the one that has more, so that they become isomorphic. We first show that this problem is NP-complete for the case where the voter matching is given (which also proves the same result for SUBELECTION ISOMORPHISM WITH

VOTER MATCHING) and next we describe how this proof can be adapted to the variant without any matchings (the variant with candidate matching is in P and was considered in the preceding section).

Theorem 5. SUBELECTION ISOMORPHISM WITH VOTER MATCHING *and* CAND.-SUBELECTION ISOMORPHISM WITH VOTER MATCHING *are* NP*-complete.*

CAND.-SUBELECTION ISOMORPHISM remains NP-complete also without the voter matching. By doubling the voters and using a few extra candidates we ensure that only the intended voter matching is possible.

Proposition 6. CAND.-SUBELECTION ISOMORPHISM *is* NP*-complete.*

4.2 Intractability of Max. Common Subelection

Perhaps the most surprising result regarding MAX. COMMON SUBELECTION is that it is NP-complete even when both matchings are given. The surprise stems from the fact that all generalizations of ELECTION ISOMORPHISM considered by Faliszewski et al. (2019) are solvable in polynomial-time in this setting. We first show this result for candidate subelections.

Theorem 7. MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS *is* NP*-complete and* W[1]*-complete with respect to the candidate set size of isomorphic candidate subelections.*

Proof. We give a reduction from the CLIQUE problem which idea is to decode the adjacency matrix of a given graph by a pair of elections with both matchings defined. Missing edges in a graph we decode as a conflict on candidates ordering within matched voters.

Formally, given an instance (G, k) of CLIQUE, we form two elections, $E_1 = (C, V_1)$ and $E_2 = (C, V_2)$, where $C = V(G)$. Since we need to provide an instance with a candidate matching, we simply specify both elections over the same candidate set. Without loss of generality, we assume that $V(G) = \{1, \ldots, n\}$. For each $x \in V(G)$ we define the neighborhood of x in G as $N(x) = \{y \in V(G) : \{x, y\} \in E(G)\}$ and the set of non-neighbors as $M(x) = V(G) \setminus \{N(x) \cup \{x\}\}$.

For each vertex $x \in V(G)$ we define two matched voters, v_x^1 in E_1 and v_x^2 in E_2, defined as follows:

$$v_x^1 \colon M(x) \succ x \succ N(x), \quad \text{and} \quad v_x^2 \colon x \succ M(x) \succ N(x).$$

We ask if E_1 and E_2 have isomorphic candidate subelections that contain at least k candidates each. Intuitively, in a solution to the problem one have to remove either x or all vertices from $M(x)$. It is a direct definition of a clique: Either x is not in a clique or all its non-neighbors are not in a clique. It is clear that the reduction can be computed in polynomial time and it remains to show its correctness.

First let us assume that G has a size-k clique. Let K be the set of this clique's vertices. We form elections E_1' and E_2' by restricting E_1 and E_2 to the candidates from K. To verify that E_1' and E_2' are isomorphic via the given matchings, let us consider an arbitrary pair of matched voters v_x^1 and v_x^2. If x is not included in K then preference orders of v_x^1 and v_x^2 restricted to K are identical. Indeed, removing even only x from the set of candidates makes v_x^1 and v_x^2 identical. Otherwise, if x is in K then $K \cap M(x) = \emptyset$ as K is a clique. Therefore, removing $M(x)$ from the set of candidates makes v_x^1 and v_x^2 identical.

For the other direction, let us assume that there are subelections E_1' and E_2' of E_1 and E_2, respectively, each with candidate set K, such that $|K| \geq k$ and E_1' and E_2' are isomorphic via the given matchings. It must be the case that the vertices from K form a clique because if K contained two vertices x and y that were not connected by an edge, then votes v_x^1 and v_x^2 would not be identical. Indeed, then we would have $y \succ x$ in v_x^1 and $x \succ y$ in v_x^2, respectively, when restricted to candidates from K. This completes the proof of NP-hardness.

To show W[1]-hardness, note that the required number of candidates in isomorphic candidate subelections is equal to the parameter k for which CLIQUE is W[1]-hard.

For membership in W[1] we show a reduction to CLIQUE with equal value of the parameters. Its idea is to create a complete graph in which vertices correspond to candidates and then remove edges based on conflicted voters. More precisely, for every two matched voters v and u and every two candidates x and y such that $x \succ_v y$ and $y \succ_u x$, we remove an edge $\{x, y\}$ from the graph. It means that both vertices corresponding to x and y cannot stay in a solution of CLIQUE. Indeed, as we cannot delete voters, the only way to resolve this conflict is to remove a candidate x or y (or both). A formal reduction is available in the full version of the paper (Faliszewski, Sornat, and Szufa 2021). □

The above reduction can be used to show strong hardness results which transfer from CLIQUE. In particular, a brute-force algorithm is essentially the best possible for exact computation. What is more, a trivial approximation algorithm which returns a constant size solution (hence the approximation ratio is $O(m)$) is also essentially optimal.

Proposition 8. MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS *has an $O^*(m^k)$-time algorithm, where k is the number of candidates in isomorphic candidate subelections and m is the number of candidates in the input (hence* XP(k)*). Moreover, assuming ETH, there is no $O^*(m^{o(k)})$-time algorithm.*

Proposition 9. MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS *has an t/c-approximation algorithm for any constant $c \geq 1$, where t is the maximum number of candidates in isomorphic candidate subelections. Moreover, approximating the problem within $t^{1-\epsilon}$ factor, for every $\epsilon > 0$, is* NP*-hard.*

All the remaining variants of MAX. COMMON CAND.-SUBELECTION also are NP-complete. The proofs either follow by applying Proposition 1 or by introducing candidates that implement a required voter matching. In the latter case, W[1]-hardness does not follow from this reduction as we introduce dummy candidates that have to be included in a solution, but their number is not a function of the CLIQUE parameter (clique size).

Proposition 10. MAX. COMMON CAND.-SUBELECTION *is* NP-*complete and so are its variants with a given candidate matching and with a given voter matching.*

Similarly to all four matching cases of the MAX. COMMON CAND.-SUBELECTION, all four matching cases of the MAX. COMMON SUBELECTION also are NP-complete.

Proposition 11. *All four matching cases of the* MAX. COMMON SUBELECTION *are* NP-*complete.*

5 Experiments

In this section we use the MAX. COMMON VOTER-SUBELECTION problem to analyze similarity between elections generated from various statistical models. While MAX. COMMON VOTER-SUBELECTION has a polynomial-time algorithm, it is too slow for our purposes. Thus we have expressed it as an integer linear program (ILP) and we were solving it using the CPLEX ILP solver. A formal ILP formulation is available in the full version of the paper (Faliszewski, Sornat, and Szufa 2021). The source code used for the experiments is available in a GitHub repository[1].

We stress that we could have used other problems from the MAX. COMMON SUBELECTION family in this section. We chose MAX. COMMON VOTER-SUBELECTION because its outcomes are particularly easy to interpret.

Our findings are similar to those of Szufa et al. (2020), but our claims of similarity between statistical cultures are stronger, whereas our dissimilarity claims are weaker.

5.1 Statistical Models of Elections

We consider the following models for generating elections:

Identity. Under the Identity model (ID), we choose a single preference order uniformly at random and then all the generated votes are equal to it.

Impartial Culture. Under Impartial Culture (IC), we generate each preference order uniformly at random.

Pólya-Eggenberger Urn Model. The Pólya-Eggenberger Urn Model (Berg 1985) is parameterized by a nonnegative value $\alpha \in [0, \infty)$, which specifies the degree of correlation between the votes (McCabe-Dansted and Slinko 2006). We start with an urn containing exactly one copy of each possible preference order; then we generate the votes one by one, by drawing a preference order from the urn (which we assume to be the generated vote) and replacing it with $\alpha \cdot m!$ duplicates, where m is the number of candidates in the election.

Mallows Model. The Mallows Model (Mallows 1957) is parameterized by a value $\phi \in [0, 1]$ and a central preference order v_c, which we choose uniformly at random. The probability of generating a preference order v is proportional to $\phi^{\mathrm{swap}(v_c,v)}$, where $\mathrm{swap}(v_c, v)$ is the minimum number of swaps of adjacent candidates needed to transform v into v_c. In our experiments we do not set the value of ϕ directly, but we use the parameterization by norm-ϕ proposed by Boehmer et al. (2021) (specifically, they used rel-$\phi = 0.5 \cdot \text{norm-}\phi$). It works as follows: For a given norm-ϕ value and a given number m of candidates in the election to be generated, we choose value ϕ so that for a generated vote v the expected value of $\mathrm{swap}(v_c, v)$ is equal to norm-ϕ times half the total number of swaps possible in preference orders over m candidates. Thus using norm-$\phi = 1$ means generating votes according to the IC model, using norm-$\phi = 0$ means using the identity model, and using norm-$\phi = 0.5$ means using a model that in a certain formal sense is exactly between these two extremes. See the work of Lu and Boutilier (2014) for an effective sampling algorithm.

1D Interval Model. The candidates and the voters are points drawn uniformly at random from a unit interval. Each voter v ranks the candidates with respect to increasing Euclidean distances of their points from that of v.

We also use the models of Walsh (2015) and Conitzer (2009) that generate single-peaked elections (it is also well-known that all 1D Interval elections are single-peaked). Given a societal axis, these models work as follows:

Walsh Model. Each single-peaked vote, for a given axis, is drawn with equal probability (we use the sampling algorithm of Walsh (2015)).

Conitzer Model. Under the Conitzer model, to generate a vote we start by choosing the top candidate uniformly at random. Then we keep on extending the vote with either of the two candidates right next to the already selected one(s) on the axis, depending on a coin toss.

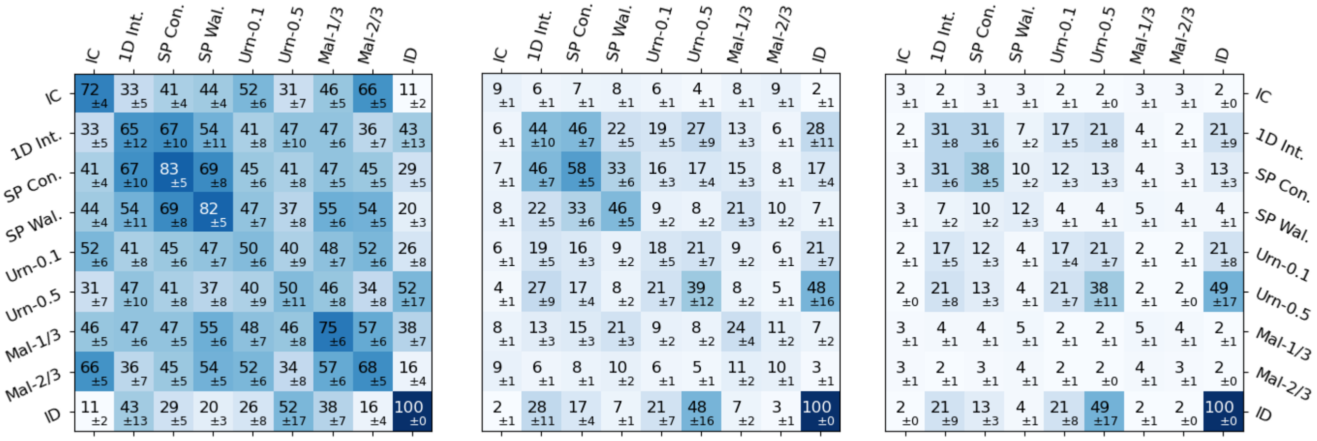
5.2 Results and Analysis

We consider elections with 4, 7, or 10 candidates and with 50 voters. For each scenario and each two of the above-described models, we have generated 1000 pairs of elections (for urn elections, we used $\alpha \in \{0.1, 0.5\}$ and for the Mallows model, we used norm-$\phi \in \{1/3, 2/3\}$). For each pair of models, we recorded the average number of voters in the maximum common voter subelections (normalized by fifty, i.e., the number of voters in the original elections), as well as the standard deviation of this value. We show our numerical results in Figure 1 (each cell corresponds to a pair of models; the number in the top-left corner is the average, and the one in the bottom-right corner is the standard deviation). Note that the matrices in Figure 1 are symmetric (results for models A and B are the same as for models B and A).

For the case with four candidates, we see that the level of similarity between elections from various models is quite high and drops sharply as the number of candidates increases. This shows that for experiments with very few candidates it is not as relevant to consider very different election models, but for more using diverse models is justified.

The above notwithstanding, some models remain similar even for 7 or 10 candidates. This is particularly visible for the case of single-peaked elections. The 1D-Interval model remains very similar to the Conitzer model, and the Walsh model is quite similar to these two for up to 7 candidates, but for 10 candidates starts to stand out (the result for 10 candidates is also visible in the experiments of Szufa et al. (2020); the similarity for fewer candidates is a new finding).

[1] https://github.com/Project-PRAGMA/Subelections-AAAI-2022



	IC	1D Int.	SP Con.	SP Wal.	Urn-0.1	Urn-0.5	Mal-1/3	Mal-2/3	ID
IC	72 ±4	33 ±5	41 ±4	44 ±4	52 ±6	31 ±7	46 ±5	66 ±5	11 ±2
1D Int.	33 ±5	65 ±12	67 ±10	54 ±11	41 ±8	47 ±10	47 ±6	36 ±7	43 ±13
SP Con.	41 ±4	67 ±10	83 ±5	69 ±8	45 ±6	41 ±8	47 ±5	45 ±5	29 ±5
SP Wal.	44 ±4	54 ±11	69 ±8	82 ±5	47 ±7	37 ±8	55 ±6	54 ±5	20 ±3
Urn-0.1	52 ±6	41 ±8	45 ±6	47 ±7	50 ±6	40 ±9	48 ±7	52 ±6	26 ±8
Urn-0.5	31 ±7	47 ±10	41 ±8	37 ±8	40 ±9	50 ±11	46 ±8	34 ±8	52 ±17
Mal-1/3	46 ±5	47 ±6	47 ±5	55 ±6	48 ±7	46 ±8	75 ±6	57 ±6	38 ±7
Mal-2/3	66 ±5	36 ±7	45 ±5	54 ±5	52 ±6	34 ±8	57 ±6	68 ±5	16 ±4
ID	11 ±2	43 ±13	29 ±5	20 ±3	26 ±8	52 ±17	38 ±7	16 ±4	100 ±0

IC	1D Int.	SP Con.	SP Wal.	Urn-0.1	Urn-0.5	Mal-1/3	Mal-2/3	ID
9 ±1	6 ±1	7 ±1	8 ±1	6 ±1	4 ±1	8 ±1	9 ±1	2 ±1
6 ±1	44 ±10	46 ±7	22 ±5	19 ±5	27 ±9	13 ±3	6 ±1	28 ±11
7 ±1	46 ±7	58 ±5	33 ±6	16 ±3	17 ±4	15 ±3	8 ±1	17 ±4
8 ±1	22 ±5	33 ±6	46 ±5	9 ±2	8 ±2	21 ±3	10 ±2	7 ±1
6 ±1	19 ±5	16 ±3	9 ±2	18 ±5	21 ±7	9 ±2	6 ±1	21 ±7
4 ±1	27 ±9	17 ±4	8 ±2	21 ±7	39 ±12	8 ±2	5 ±1	48 ±16
8 ±1	13 ±3	15 ±3	21 ±3	9 ±2	8 ±2	24 ±4	11 ±2	7 ±2
9 ±1	6 ±1	8 ±1	10 ±2	6 ±1	5 ±1	11 ±2	10 ±1	3 ±1
2 ±1	28 ±11	17 ±4	7 ±1	21 ±7	48 ±16	7 ±2	3 ±1	100 ±0

IC	1D Int.	SP Con.	SP Wal.	Urn-0.1	Urn-0.5	Mal-1/3	Mal-2/3	ID	
3 ±1	2 ±1	3 ±1	3 ±1	2 ±1	2 ±0	3 ±1	3 ±1	2 ±0	IC
2 ±1	31 ±8	31 ±6	7 ±2	17 ±5	21 ±8	4 ±1	2 ±1	21 ±9	1D Int.
3 ±1	31 ±6	38 ±5	10 ±2	12 ±3	13 ±3	4 ±1	3 ±1	13 ±3	SP Con.
3 ±1	7 ±2	10 ±2	12 ±3	4 ±1	4 ±1	5 ±1	4 ±1	4 ±1	SP Wal.
2 ±1	17 ±5	12 ±3	4 ±1	17 ±4	21 ±7	2 ±1	2 ±1	21 ±8	Urn-0.1
2 ±0	21 ±8	13 ±3	4 ±1	21 ±7	38 ±11	2 ±1	2 ±0	49 ±17	Urn-0.5
3 ±1	4 ±1	4 ±1	5 ±1	2 ±1	2 ±1	5 ±1	4 ±1	2 ±1	Mal-1/3
3 ±1	2 ±1	3 ±1	4 ±1	2 ±1	2 ±0	4 ±1	3 ±1	2 ±0	Mal-2/3
2 ±0	21 ±9	13 ±3	4 ±1	21 ±8	49 ±17	2 ±1	2 ±0	100 ±0	ID

Figure 1: The large numbers denote the rounded % of matched votes for MAX. COMMON VOTER-SUBELECTION. The small numbers denote the rounded standard deviation. In the left/center/right matrix there are results for 4/7/10 candidates.

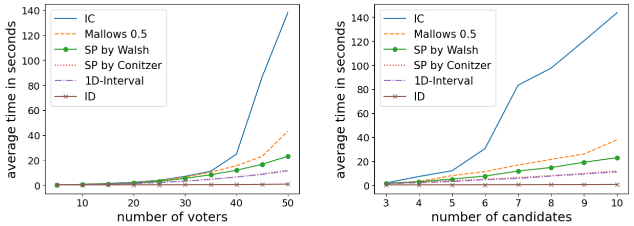


Figure 2: Average time needed to find the maximum common voter subelections. With the fixed number of candidates (left), and fixed number of voters (right).

We also note that the urn models remain relatively similar to each other (and to the 1D-Interval and Conitzer models) for all numbers of candidates, but this is not the case for the Mallows models. One explanation for this is that the urn model proceeds by copying some of the votes already present in the election, whereas the Mallows model generates votes by perturbing the central one. The former leads to more identical votes in an election. Indeed, to verify this it suffices to consider the "ID" column (or row) of the matrix: The similarity to the identity elections simply shows how often the most frequent vote appears in elections from a given model. For 10 candidates, urn elections with $\alpha \in \{0.1, 0.5\}$ have, on average, $21 \pm 8\%$ and $49 \pm 17\%$ identical votes, respectively. For Mallows elections, this value drops to around 2% (in our setting, this means 1 or 2 voters, on average).

Finally, we consider the diagonals of the matrices in Figure 1, which show self-similarity of our models. Intuitively, the larger are these values, the fewer elections of a given type one needs in an experiment. Single-peaked elections stand out here for all numbers of candidates, whereas the urn models become more prominent for larger candidate sets.

We have also analyzed the average running time that CPLEX needed to find the maximum common voter subelections[2]. We focus on IC, Identity, Walsh model, Conitzer model, Mallows model with norm-$\phi = 0.5$, and 1D-Interval. First, we generated 500 pairs of elections from each model with 10 candidates and $5, 10, \dots, 45, 50$ voters, and calculated the average time needed to find the maximum common voter subelections. Second, we fixed the number of voters to 50 and generated elections with $3, 4, \dots, 9, 10$ candidates. The results are presented in Figure 2. As we increase the number of voters, the time seems to increase exponentially. We observe large differences between the models, with IC being by far the slowest. Conitzer model and Walsh model are significantly different from one another, even though both of them generate single-peaked elections. Moreover, the fact that 1D-Interval and Conitzer models need on average the same amount of time confirms their similarity.

6 Conclusions

We have shown that variants of ELECTION ISOMORPHISM that are based on considering subelections are largely intractable but, nonetheless, some of them can be solved in polynomial-time. Indeed, we have used the polynomial-time solvable MAXIMUM COMMON CANDIDATE SUBELECTION problem to analyze similarity between various different models of generating random elections.

In Section 4 we have classified variants of the problem as either belonging to P or being NP-complete (and some being W[1]-hard). For some variants of our problems we have shown strong inapproximability results and matching approximation algorithms. However, it would also be desirable to consider the parameterized complexity of these problem for the remaining variants.

It would be interesting to consider variants of our problems where instead of requiring identical preference orders among matched voters, we might ask for similar ones (e.g., within a given swap distance). Another interesting research direction is to repeat our experiments on real-life elections.

[2] We ran CPLEX on a single thread (Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GH) of a 448 thread machine with 6TB of RAM.

Acknowledgments

Stanisław Szufa was supported by the National Science Centre, Poland (NCN) grant No 2018/29/N/ST6/01303. 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).

References

Arvind, V.; Köbler, J.; Kuhnert, S.; and Vasudev, Y. 2012. Approximate Graph Isomorphism. In *Proceedings of MFCS-12*, 100–111.

Babai, L.; Dawar, A.; Schweitzer, P.; and Torán, J. 2015. The Graph Isomorphism Problem (Dagstuhl Seminar 15511). *Dagstuhl Reports*, 5(12): 1–17.

Ballester, M.; and Haeringer, G. 2011. A Characterization of the Single-Peaked Domain. *Social Choice and Welfare*, 36(2): 305–322.

Bartholdi, J., III; and Trick, M. 1986. Stable Matching with Preferences Derived from a Psychological Model. *Operations Research Letters*, 5(4): 165–169.

Berg, S. 1985. Paradox of Voting Under an Urn Model: The Effect of Homogeneity. *Public Choice*, 47(2): 377–387.

Black, D. 1958. *The Theory of Committees and Elections*. Cambridge University Press.

Boehmer, N.; Bredereck, R.; Faliszewski, P.; Niedermeier, R.; and Szufa, S. 2021. Putting a Compass on the Map of Elections. In *Proceedings of IJCAI-21*, 59–65.

Bredereck, R.; Chen, J.; and Woeginger, G. 2013. A Characterization of the Single-Crossing Domain. *Social Choice and Welfare*, 41(4): 989–998.

Chen, J.; Huang, X.; Kanj, I. A.; and Xia, G. 2006. Strong Computational Cower Bounds via Parameterized Complexity. *Journal of Computer and System Sciences*, 72(8): 1346–1367.

Conitzer, V. 2009. Eliciting Single-Peaked Preferences Using Comparison Queries. *Journal of Artificial Intelligence Research*, 35: 161–191.

Cook, S. 1971. The Complexity of Theorem-Proving Procedures. In *Proceedings of STOC-71*, 151–158.

Cygan, M.; Fomin, F.; Kowalik, Ł.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer.

Doignon, J.; and Falmagne, J. 1994. A Polynomial Time Algorithm for Unidimensional Unfolding Representations. *Journal of Algorithms*, 16(2): 218–233.

Downey, R. G.; and Fellows, M. R. 1995. Fixed-Parameter Tractability and Completeness II: On Completeness for W[1]. *Theoretical Computer Science*, 141(1-2): 109–131.

Elkind, E.; Faliszewski, P.; and Slinko, A. 2012. Clone Structures in Voters' Preferences. In *Proceedings of EC-12*, 496–513.

Escoffier, B.; Lang, J.; and Öztürk, M. 2008. Single-Peaked Consistency and its Complexity. In *Proceedings of ECAI-08*, 366–370.

Faliszewski, P.; Skowron, P.; Slinko, A.; Szufa, S.; and Talmon, N. 2019. How Similar Are Two Elections? In *Proceedings of AAAI-19*, 1909–1916.

Faliszewski, P.; Sornat, K.; and Szufa, S. 2021. The Complexity of Subelection Isomorphism Problems. arXiv:2105.11923.

Garey, M.; and Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

Grohe, M.; Rattan, G.; and Woeginger, G. 2018. Graph Similarity and Approximate Isomorphism. In *Proceedings of MFCS-18*, 20:1–20:16.

Lu, T.; and Boutilier, C. 2014. Effective Sampling and Learning for Mallows Models with Pairwise-Preference Data. *Journal of Machine Learning Research*, 15(1): 3783–3829.

Mallows, C. 1957. Non-Null Ranking Models. *Biometrica*, 44: 114–130.

McCabe-Dansted, J.; and Slinko, A. 2006. Exploratory Analysis of Similarities Between Social Choice Rules. *Group Decision and Negotiation*, 15: 77–107.

Mirrlees, J. 1971. An Exploration in the Theory of Optimal Income Taxation. *Review of Economic Studies*, 38: 175–208.

Papadimitriou, C. 1994. *Computational Complexity*. Addison-Wesley.

Raymond, J.; and Willett, P. 2002. Maximum Common Subgraph Isomorphism Algorithms for the Matching of Chemical Structures. *Journal of Computer-Aided Molecular Design*, 16(7): 521–533.

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

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

Vayer, T.; Redko, I.; Flamary, R.; and Courty, N. 2020. CO-Optimal Transport. In *Proceedings of NeurIPS-20*.

Walsh, T. 2015. Generating Single Peaked Votes. arXiv:1503.02766.