

# The Complexity of Subelection Isomorphism Problems

**Piotr Faliszewski**  
**Krzysztof Sornat**  
*AGH University, Kraków, Poland,*

FALISZEW@AGH.EDU.PL  
SORNAT@AGH.EDU.PL

**Stanisław Szufa**  
*AGH University, Kraków, Poland*  
*CNRS, LAMSADE, Université Paris Dauphine-PSL, France*

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, Dawar, Schweitzer, & Torán, 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 et al. (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 first taken by Faliszewski et al. (2019), who introduced the isomorphic swap distance between elections: Given two elections, their isomorphic swap distance is the number of swaps of adjacent candidates that we need to perform in one of them to make it isomorphic to the other one (they also considered a related notion based on the Spearman distance between votes). As they showed that computing their distances is computationally challenging, Vayer et al. (2020) sought practical ways of circumventing this intractability, and Szufa et al. (2020) proposed a different, polynomial-time computable metric, which, however, is less precise (while it ensures that isomorphic elections are at distance zero, it does not guarantee that distance zero means that

elections are isomorphic). Using this new distance, Szufa et al. (2020) and Boehmer et al. (2021) developed the *map of elections* framework, that is, a technique for presenting similarities between elections graphically, later studied in many of their follow-up works (Boehmer, Faliszewski, Niedermeier, Szufa, & Waś, 2022; Boehmer, Bredereck, Elkind, Faliszewski, & Szufa, 2022; Boehmer, Cai, Faliszewski, Fan, Janeczko, Kaczmarczyk, & Waś, 2023; Faliszewski, Kaczmarczyk, Sornat, Szufa, & Waś, 2023).

Our goal is to explore the other way of relaxing the notion of election isomorphism, focused on seeking isomorphic substructures. Specifically, we consider the SUBELECTION ISOMORPHISM and MAXIMUM COMMON SUBELECTION families of problems.<sup>1</sup> 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. 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 subelections (Ballester & Haeringer, 2011; Bredereck, Chen, & 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 subelections of constant size and, hence, such elections can be recognized efficiently; indeed, there are very fast algorithms for these tasks (Bartholdi III & Trick, 1986; Escoffier, Lang, & Öztürk, 2008; Elkind, Faliszewski, & Slinko, 2012). Our results show that characterizations with subelections of non-constant size 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), discussed above. The main difference, as compared to the work of Faliszewski et al. (2019), is that some of our problems—including practically useful ones—are polynomial-time solvable, and the main difference from the distance proposed by Szufa et al. (2020) is that using the MAXIMUM COMMON SUBELECTION problem we can give much stronger indication of similarity between elections (but our indication of dissimilarity is much weaker).

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 and some real-life elections. These results confirm some findings observed by Szufa et al. (2020) and Boehmer

---

1. Problems from graph theory that inspired ours are SUBGRAPH ISOMORPHISM (Karp, 1972) and MAXIMUM COMMON SUBGRAPH (see, e.g., the work of Kann (1992) for an early reference). In the former, we ask if a given smaller graph is isomorphic to a subgraph of a given larger one. In the latter, we ask for the largest graph that is isomorphic to subgraphs of two given ones. The literature also includes many variants of these problems and, as an example, we mention the MAXIMUM COMMON EDGE SUBGRAPH problem of Bokhari (1981), where we seek a common subgraph of two given ones that has as many edges as possible.

Problem	no matching	voter-matching	cand.-matching	both matchings
ELECTION ISOMORPHISM	P	P	P	P
SUBELECTION ISOM.	W[1]-h. [Thm. 4.2]	NP-com. [Thm. 4.4]	P [Thm. 4.1]	P [Thm. 4.1]
CAND.-SUBELECTION ISOM.	NP-com. [Prop. 4.5]	NP-com. [Thm. 4.4]	P [Thm. 4.1]	P [Thm. 4.1]
VOTER-SUBELECTION ISOM.	P [Thm. 4.1]	P [Thm. 4.1]	P [Thm. 4.1]	P [Thm. 4.1]
MAX. COMMON SUBEL.	W[1]-h. [Prop. 4.10]	NP-com. [Prop. 4.10]	NP-com. [Prop. 4.10]	NP-com. [Prop. 4.10]
MAX. COMMON CAND.-SUBEL.	NP-com. [Prop. 4.9]	NP-com. [Prop. 4.9]	NP-com. [Prop. 4.9]	W[1]-com. [Thm. 4.6]
MAX. COMMON VOTER-SUBEL.	P [Thm. 4.1]	P [Thm. 4.1]	P [Thm. 4.1]	P [Thm. 4.1]

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. Problems indicated as W[1]-hard problems are also NP-hard.

et al. (2021) in their maps of elections (such as the similarity between various ways of generating single-peaked elections) and give a new perspective on some of these statistical cultures and real-life elections.

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. This way we follow Faliszewski et al. (2019), who considered the complexity of computing their distances for the cases where such matchings are given. While one would expect that having the 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), who have shown that isomorphic swap distance is polynomial-time computable in this setting. 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,  $v: c_i \succ c_j$ ) 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  $v: c_1 \succ c_2 \succ \dots \succ c_m$  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  $\triangleleft$  be a linear order over  $C$  (referred to as the societal axis). We say that a vote  $v$  is single-peaked with respect to  $\triangleleft$  if for each  $k \in [m]$ , the top  $k$  candidates in  $v$  form an interval within  $\triangleleft$ . 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  $\sigma: C \rightarrow D$  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  $\sigma: C_1 \rightarrow C_2$  and a permutation  $\pi \in S_{|V_1|}$  such that  $(\sigma(C_1), \sigma(\pi(V_1))) = (C_2, V_2)$ . We refer to  $\sigma$  as the candidate matching and to  $\pi$  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 & 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 the textbook of 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 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:

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

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  $t$ .<sup>2</sup>

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  $\sigma$  that matches (some of) the candidates in one election to (some of) those in the other. Then we ask for an isomorphism between respective subelections that agrees with  $\sigma$ . In particular, this means that none of the unmatched candidates remain in the considered subelections. Consequently, in case of SUBELECTION ISOMORPHISM with candidate matching and its variants, all the candidates

---

2. It would also be natural to include two numbers in an instance,  $t_c$  and  $t_v$ , and ask for  $E'_1$  and  $E'_2$  that both have at least  $t_c$  candidates and  $t_v$  voters. The choice between these two formulations—with number  $t$  or with numbers  $t_c$  and  $t_v$ —is somewhat arbitrary, in the sense that computational complexity for both variants is the same. Indeed, in our hardness proofs we always have that if there is a solution then it has certain numbers of candidates and voters, and our polynomial-time algorithms only regard variants of MAX. COMMON VOTER SUBELECTION, where we cannot delete candidates and the two variants are equivalent.

from the smaller election must be matched to those in the larger one. For variants of our problems where each candidate from one election is matched to some candidate from the other, a convenient interpretation of having a candidate matching 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  $\sigma$  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  $v_1: x \succ w, v_2: w \succ x$ , and  $v_3: w \succ x$ , 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  $\sigma$ , of size  $2 \cdot 2 = 4$ .

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. The sought-after isomorphism must respect this matching; again, this means that we can disregard the unmatched voters and, hence, for SUBELECTION ISOMORPHISM and its variants, each voter from the smaller election must be matched to some voter from the larger one.

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 3.1.** 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.

*Proof.* We are given a problem  $M$  which is a variant of MAX. COMMON SUBELECTION, and problem  $S$ , which is an analogous variant of SUBELECTION ISOMORPHISM (so if the former only allows deleting candidates, then so does the latter, etc.). We want to show that  $S$  many-one reduces to  $M$  in polynomial time. Let  $I_S = (E_1, E_2)$  be an instance of  $S$ , where  $E_1$  is the smaller election. We form an instance  $I_M = (E_1, E_2, t)$  of  $M$ , which uses the very same elections and where  $t$  is set to be the size of  $E_1$ . This means that in  $I_M$  we cannot perform any operations on election  $E_1$ , because that would decrease its size below  $t$ . Hence, we can only perform operations on  $E_2$ , so the situation is the same as in the  $S$  problem and in the  $I_S$  instance.  $\square$

## 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 algorithms for all the variants of SUBELECTION ISOMORPHISM.

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 4.1.** *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 all cases with candidate matchings.*

*Proof.* We first give an algorithm for MAX. COMMON VOTER-SUBELECTION. Let  $E = (C, V)$  and  $F = (D, U)$  be our input elections and let  $t$  be the desired size of their isomorphic subelections. Since we are looking for a voter subelection, without loss of generality we may assume that  $|C| = |D|$  (and we write  $m$  to denote the number of candidates in each set). For each pair of voters  $v \in V$  and  $u \in U$  we perform the following algorithm:

1. Denoting the preference orders of  $v$  and  $u$  as  $c_1 \succ_v c_2 \succ_v \dots \succ_v c_m$  and  $d_1 \succ_u d_2 \succ_u \dots \succ_u d_m$ , respectively, we form a bijection  $\sigma: C \rightarrow D$  such that for each  $c_i \in C$  we have  $\sigma(c_i) = d_i$ .
2. We form a bipartite graph where the voters from  $V$  form one set of vertices, the voters from  $U$  form the other set of vertices, and there is an edge between voters  $v' \in V$  and  $u' \in U$  if  $\sigma(v') = u'$ .
3. We compute the maximum cardinality matching in this graph, using, e.g., one of the classic polynomial-time algorithms; see the overview of Ahuja et al. (1993) or the textbook of Cormen et al. (2001). Then, we form subelections that consist of the matched voters. We accept if their size is at least  $t$ .

If the algorithm does not accept for any choice of  $v$  and  $u$ , we reject.

Very similar algorithms also work for the variants of MAX. COMMON VOTER-SUBELECTION with either one or both of the matchings: If we are given the candidate matching, then we can omit the first step in the enumerated algorithm above, and if we are given a voter matching then instead of trying all pairs of voters  $v$  and  $u$  it suffices to try all voters from the first election and obtain the other one via the matching. Analogous algorithms also work for VOTER-SUBELECTION ISOMORPHISM (for all four matching cases) and for all the other variants of SUBELECTION ISOMORPHISM, provided that the candidate matching is given.  $\square$

#### 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 4.2.** *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,  $\alpha_H$  and  $\beta_H$ , 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:

$$\begin{aligned} v_e^1: & x \succ y \succ \alpha_H \succ \beta_H \succ V(H) \setminus \{x, y\}, \\ v_e^2: & x \succ y \succ \beta_H \succ \alpha_H \succ V(H) \setminus \{x, y\}, \\ v_e^3: & y \succ x \succ \alpha_H \succ \beta_H \succ V(H) \setminus \{x, y\}, \\ v_e^4: & y \succ x \succ \beta_H \succ \alpha_H \succ V(H) \setminus \{x, y\}. \end{aligned}$$

Note that the elements from the set  $V(H) \setminus \{x, y\}$  are always ranked in the same order (as per our convention, including a candidate set in a preference order means listing its members according to some arbitrary, but fixed, global order). 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 its correctness. Without loss of generality, we assume that  $k \geq 3$ .

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  $\alpha_G$  and  $\beta_G$ . To see why this is so, consider the following two cases:

1. If  $E'$  contained exactly one of  $\alpha_G, \beta_G$ , 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  $\alpha_G$  nor  $\beta_G$  then every vote in  $E'$  would rank some vertex candidates  $z$  and  $w$  on positions three and four (to be able to match  $\alpha_K$  and  $\beta_K$  to them). In particular, this would mean that  $z$  and  $w$  would never appear on top two positions in  $E'$  and, hence, that  $E'$  would not include votes corresponding to edges involving vertices  $z$  and  $w$ . 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  $\alpha_G$  and  $\beta_G$  are included in  $E'$ . Moreover  $\alpha_G$  and  $\beta_G$  are matched with  $\alpha_K$  and  $\beta_K$ . To see why this is the case, let us consider the following two cases: (a) If  $E'$  contains a vote where  $\alpha_G$  appears on position three then it must also include a vote where  $\alpha_G$  appears on position four (this holds because in  $E_k$  every candidate that appears on position three in some vote also appears on position four in some other vote). However,  $\alpha_G$  appears on position four in some vote from  $E'$  only if  $\beta_G$  appears on position three in this vote. Hence, this vote witnesses that  $\alpha_G$  and  $\beta_G$  must be matched to  $\alpha_K$  and  $\beta_K$  (because  $\alpha_K$  and  $\beta_K$  are the only candidates that appear on positions three and four in  $E_K$ ). Analogous reasoning applies to the case where  $\beta_G$  appears on position three in some vote from  $E'$ . (b) If the preceding case does not apply, then all votes in  $E'$  rank  $\alpha_G$  and  $\beta_G$  on the top two positions.



However, this means that  $E'$  is not isomorphic to  $E_K$  because in the latter election there are no two candidates that are always ranked on the top two positions.

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  $\alpha_G$  or  $\beta_G$ —that appeared in all the votes within the top four positions and in some vote within the 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). The number of voters in  $E_K$  is  $4\binom{k}{2}$ , and the number of distinct corresponding edges from  $G$  is  $\binom{k}{2}$ . 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  $\alpha_G$  and  $\beta_G$ ) is between  $k$  and  $2\binom{k}{2}$ . However, the number of candidates in  $E_K$  except  $\alpha_K$  and  $\beta_K$  is  $k$ , therefore we conclude that the 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.  $\square$

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 4.3.** *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 the problem is in XP for the parameter  $k$ ). Moreover, assuming ETH, there is no  $O^*(m^{o(k)})$ -time algorithm.*

*Proof.* The XP algorithm is a straightforward brute-force approach. For a fixed order of candidates in the smaller election, we create a matching with some  $k$  candidates in the larger election. We try at most  $m(m-1)\dots(m-k+1) \leq m^k$  cases. For a given matching, we solve a problem SUBELECTION ISOMORPHISM WITH CANDIDATE MATCHING using a polynomial-time algorithm from Theorem 4.1. Hence the total running time is  $O^*(m^k)$ .

For the hardness part, recall that ETH implies that there is no  $|V|^{o(K)}$ -time algorithm for finding a size- $K$  clique on a graph with vertex set  $V$  (Chen et al., 2006). In our reduction in Theorem 4.2, we have that the number of candidates in the larger election is  $m = |V| + 2$  and the number of candidates in the smaller election is  $k = K + 2$ . Therefore, using an  $O^*(m^{o(k)})$ -time algorithm for SUBELECTION ISOMORPHISM we could solve CLIQUE in time  $m^{o(k)} \cdot \text{poly}(k^3 + m^3) = (|V| + 2)^{o(K+2)} \cdot \text{poly}(|V|) = |V|^{o(K)}$ . This would contradict ETH.  $\square$

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 & 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 et al., 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 & 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 4.4.** SUBELECTION ISOMORPHISM WITH VOTER MATCHING and CAND.-SUBELECTION ISOMORPHISM WITH VOTER MATCHING are NP-complete.

*Proof.* We note that each instance of CAND.-SUBELECTION ISOMORPHISM WITH VOTER MATCHING is also an instance of SUBELECTION ISOMORPHISM WITH VOTER MATCHING. Consequently, it suffices to show NP-hardness of the former (as per our convention, we omit simple NP-membership proofs). We do so by giving a reduction from EXACT COVER BY 3-SETS (X3C). An instance of X3C consists of a set  $X = \{x_1, \dots, x_m\}$  of elements and a family  $\mathcal{S} = \{S_1, \dots, S_n\}$  of three-element subsets of  $X$ . We ask if  $\mathcal{S}$  contains a subfamily  $\mathcal{S}'$  such that each element from  $X$  belongs to exactly one set from  $\mathcal{S}'$ . Given such an instance, we form two elections,  $E_1$  and  $E_2$ , as follows.

Election  $E_1$  will be our smaller election and  $E_2$  will be the larger one. We let  $X$  be the candidate set for election  $E_1$ , whereas to form the candidate set of  $E_2$  we proceed as follows. For each set  $S_t = \{x_i, x_j, x_k\} \in \mathcal{S}$ , we introduce candidates  $s_{i,t}$ ,  $s_{j,t}$ , and  $s_{k,t}$ . Intuitively, if some candidate  $s_{u,t}$  remains in a subelection isomorphic to  $E_1$ , then we will interpret this fact as saying that element  $x_u$  is covered by set  $S_t$ ; this will, of course, require introducing appropriate consistency gadgets to ensure that  $S_t$  also covers its other members. For each  $i \in [m]$ , we let  $P_i$  be the set of all the candidates of the form  $s_{i,t}$ , where  $t$  belongs to  $[n]$  (in other words, each  $P_i$  contains the candidates from  $E_2$  that are associated with element  $x_i$ ).

**Example 4.1.** Let  $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$  and  $\mathcal{S} = \{S_1, \dots, S_4\}$ , where:

$$S_1 = \{x_1, x_2, x_5\}, \quad S_2 = \{x_1, x_3, x_6\}, \quad S_3 = \{x_2, x_3, x_5\}, \quad S_4 = \{x_3, x_4, x_6\}.$$

Then:

$$\begin{aligned} P_1 &= \{s_{1,1}, s_{1,2}\}, & P_2 &= \{s_{2,1}, s_{2,3}\}, & P_3 &= \{s_{3,2}, s_{3,3}, s_{3,4}\}, \\ P_4 &= \{s_{4,4}\}, & P_5 &= \{s_{5,1}, s_{5,3}\}, & P_6 &= \{s_{6,2}, s_{6,4}\}. \end{aligned}$$

We denote the voter collection of election  $E_1$  as  $V = (v, v', v_1, v'_1, \dots, v_n, v'_n)$  and the voter collection of  $E_2$  as  $U = (u, u', u_1, u'_1, \dots, u_n, u'_n)$ . Voters  $v$  and  $v'$  are matched to voters  $u$  and  $u'$ , respectively, and for each  $i \in [2n]$ ,  $v_i$  is matched to  $u_i$  and  $v'_i$  is matched to  $u'_i$ . The preference orders of the first two pairs of voters are:

$$\begin{aligned} v: x_1 \succ x_2 \succ \dots \succ x_m, & & u: P_1 \succ P_2 \succ \dots \succ P_m, \\ v': x_1 \succ x_2 \succ \dots \succ x_m, & & u': \overleftarrow{P}_1 \succ \overleftarrow{P}_2 \succ \dots \succ \overleftarrow{P}_m. \end{aligned}$$

Next, for each  $t \in [n]$  such that  $S_t = \{x_i, x_j, x_k\}$ ,  $i < j < k$ , we let the preference orders of  $v_t, v'_t$  and their counterparts from  $U$  be as follows (by writing “...” in the votes from  $E_1$  we mean listing the candidates from  $X \setminus \{x_i, x_j, x_k\}$  in the order of increasing indices, and for the voters from  $E_2$  by “...” we mean order  $P_1 \succ P_2 \succ \dots \succ P_m$  with  $P_i, P_j$ , and  $P_k$  removed):

$$\begin{aligned} v_t: & x_i \succ x_j \succ x_k \succ \dots, \\ u_t: & s_{i,t} \succ s_{j,t} \succ s_{k,t} \succ P_i \setminus \{s_{i,t}\} \succ P_j \setminus \{s_{j,t}\} \succ P_k \setminus \{s_{k,t}\} \succ \dots, \\ v'_t: & x_k \succ x_j \succ x_i \succ \dots, \\ u'_t: & s_{k,t} \succ s_{j,t} \succ s_{i,t} \succ P_k \setminus \{s_{k,t}\} \succ P_j \setminus \{s_{j,t}\} \succ P_i \setminus \{s_{i,t}\} \succ \dots. \end{aligned}$$

This finishes our construction. It is clear that it is polynomial-time computable and it remains to show that it is correct.

Let us assume that we have a *yes*-instance of X3C, that is, there is a family  $\mathcal{S}'$  of sets from  $\mathcal{S}$  such that each element from  $X$  belongs to exactly one set from  $\mathcal{S}'$ . We form a subelection  $E'$  of  $E_2$  by deleting all the candidates  $s_{i,t}$  except for those for whom set  $S_t$  belongs to  $\mathcal{S}'$ . Then, let  $\sigma$  be a function such that for each  $x_i \in X$  we have  $\sigma(x_i) = s_{i,t}$ , where  $S_t$  is the set from  $\mathcal{S}'$  that contains  $x_i$ . Together with our voter matching,  $\sigma$  witnesses that  $E_1$  and  $E'$  are isomorphic.

For the other direction, let us assume that  $E_2$  has a subelection  $E'$  that is isomorphic to  $E_1$  and let  $C'$  be its candidate set. First, we claim that for each  $i \in [m]$  exactly one candidate from  $P_i$  is included in  $C'$ . Indeed, if it were not the case, then  $u$  and  $u'$  would not have identical preference orders, as required by the fact that they are matched to  $v$  and  $v'$ . Second, we note that for each  $i \in [m]$  the candidate matching that witnesses our isomorphism must match  $x_i$  with the single candidate in  $P_i \cap C'$ . Finally, we claim that if some candidate  $s_{i,t}$  is included in  $C'$ , where  $S_t = \{x_i, x_j, x_k\}$ ,  $i < j < k$ , then candidates  $s_{j,t}$  and  $s_{k,t}$  are included in  $C'$  as well. Indeed, if  $s_{j,t}$  were not included in  $C'$ , then  $u_t$  and  $u'_t$  would rank the members of  $P_i \cap C'$  and the members of  $P_j \cap C'$  in the same order, whereas  $v_j$  and  $v'_j$  rank  $x_i$  and  $x_j$  in opposite orders (and, by the second observation,  $x_i$  and  $x_j$  are matched to the member of  $P_i \cap C'$  and  $P_j \cap C'$ , respectively). The same argument applies to  $x_{k,t}$ . We say that a set  $S_t = \{x_i, x_j, x_k\} \in \mathcal{S}$  is selected by  $C'$  if the candidates  $s_{i,t}$ ,  $s_{j,t}$ , and  $s_{k,t}$  belong to  $C'$ . By the above reasoning, we see that exactly  $n/3$  sets are selected and that they form an exact cover of  $X$ .  $\square$

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 4.5.** CAND.-SUBELECTION ISOMORPHISM is NP-complete.

*Proof.* We give a reduction from CAND. SUBELECTION ISOMORPHISM WITH VOTER MATCHING. Let the input instance be  $(E_1, E_2)$ , where the smaller election,  $E_1$ , has voter collection  $(v_1, \dots, v_n)$  and the larger election,  $E_2$ , has voter collection  $(u_1, \dots, u_n)$ . Further, for each  $i \in [n]$  voter  $v_i$  is matched with voter  $u_i$ . We form elections  $E'_1$  and  $E'_2$  in the following way. The candidate set of  $E'_1$  is the same as that of  $E_1$  except that it also includes candidates from the set  $D = \{d_1, \dots, d_{2n}\}$ . Similarly,  $E'_2$  contains the same candidates as

$E_2$  plus the candidates from the set  $F = \{f_1, \dots, f_{2n}\}$ . The voter collections of  $E'_1$  and  $E'_2$  are, respectively,  $(v'_1, v''_1, \dots, v'_n, v''_n)$  and  $(u'_1, u''_1, \dots, u'_n, u''_n)$ . For each  $i \in [n]$  these voters have the following preference orders (by writing  $[v_i]$  or  $[u_i]$  in a preference order we mean inserting the preference order of voter  $v_i$  or  $u_i$  in a given place):

$$\begin{aligned} v'_i &: d_{2i-1} \succ d_1 \succ \dots \succ d_{2n} \succ [v_i], \\ u'_i &: f_{2i-1} \succ f_1 \succ \dots \succ f_{2n} \succ [u_i], \\ v''_i &: d_{2i} \succ d_1 \succ \dots \succ d_{2n} \succ [v_i], \\ u''_i &: f_{2i} \succ f_1 \succ \dots \succ f_{2n} \succ [u_i]. \end{aligned}$$

We claim that  $E'_1$  is isomorphic to a candidates subelection of  $E'_2$  if and only if  $E_1$  is isomorphic to a candidate subelection of  $E_2$  with the given voter matching. In one direction this is clear: If  $E_1$  is isomorphic to a subelection of  $E_2$  with a given voter matching, then it suffices to use the same voter matching for the case of  $E'_1$  and  $E'_2$ , and the same candidate matching, extended with matching each candidate  $d_i$  to  $f_i$ .

Next, let us assume that  $E'_1$  is isomorphic to some subelection  $E'$  of  $E'_2$ . By a simple counting argument, we note that  $E'$  must contain some candidates not in  $F$ . Further, we also note that it must contain all members of  $F$ . Indeed, each voter in  $E'_1$  has a different candidate on top and this would not be the case in  $E'$  if it did not include all members of  $F$  (if  $E'$  did not include any members of  $F$  then this would hold for each two votes  $u'_i$  and  $u''_i$ , and if  $E'$  contained some members of  $F$  but not all of them, then this would hold because each voter in  $E'$  would rank some member of  $F$  on top, but there would be fewer members of  $F$  than voters in the election).

As a consequence, every candidate matching  $\sigma$  that witnesses isomorphism between  $E'_1$  and  $E'$  matches some member of  $D$  to some member of  $F$ . Further, we claim that for each  $i \in [2n]$ ,  $\sigma(d_i) = f_i$ . For the sake of contradiction, let us assume that this is not the case and consider some  $i \in [2n - 1]$  for which there are  $j$  and  $k$  such that  $\sigma(d_i) = f_j$ ,  $\sigma(d_{i+1}) = f_k$  and  $j > k$  (such  $i, j, k$  must exist under our assumption). However, in  $E'_1$ , all but one voter rank  $d_i$  ahead of  $d_{i+1}$ , whereas in  $E'$  all but one voter rank  $\sigma(d_{i+1})$  ahead of  $\sigma(d_i)$ . Thus  $\sigma$  cannot witness isomorphism between  $E'_1$  and  $E'$ .

Finally, since for each  $i \in [2n]$  we have that  $d_i$  is matched to  $f_i$ , it also must be the case that for each  $j \in [n]$  voters  $v'_j$  and  $v''_j$  are matched to  $u'_j$  and  $u''_j$ , respectively (indeed,  $v'_j$  is the only voter who ranks  $d_{2j-1}$  on top, and  $u'_j$  is the only voter who ranks  $f_{2j-1}$  on top; the same argument works for the other pair of voters). As a consequence, we have that  $E_1$  is isomorphic to a subelection of  $E_2$  under the voter matching that for each  $i \in [n]$  matches  $v_i$  to  $u_i$ .  $\square$

## 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 4.6.** 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, where the idea is to encode the adjacency matrix of a given graph by a pair of elections with both matchings defined. We encode the missing edges in the graph as a conflict on the candidate 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 candidate matching, we simply specify both elections over the same candidate set. Without loss of generality, we assume that  $V(G) = \{1, \dots, n\}$ . 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,  $v_x$  in  $E_1$  and  $u_x$  in  $E_2$ , with the following preference orders:

$$\begin{aligned} v_x &: M(x) \succ x \succ N(x), \\ u_x &: x \succ M(x) \succ N(x). \end{aligned}$$

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, for each vertex  $x$  one has 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 nonneighbors 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  $v_x$  and  $u_x$ . If  $x$  is not included in  $K$  then the preference orders of  $v_x$  and  $u_x$  restricted to  $K$  are identical. Indeed, removing even only  $x$  from the set of candidates makes  $v_x$  and  $u_x$  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  $v_x$  and  $u_x$  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  $v_x$  and  $u_x$  would not be identical. Indeed, we would have  $y \succ x$  in  $v_x$  and  $x \succ y$  in  $u_x$ , 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.

To show membership of MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS in W[1], let us now give a reduction to CLIQUE with an equal value of the parameters. Let  $E_1 = (C, V_1)$  and  $E_2 = (C, V_2)$  be our input elections and let  $k$  be the number of candidates in maximum isomorphic candidate subelections (since we are in the “with candidate matching” regime, we take the candidate sets to be equal). Let  $m = |C|$ ,  $n = |V_1| = |V_2|$  (since we cannot remove the voters).

We create an instance  $(G, k)$  of CLIQUE as follows. We define  $G$  as having vertices corresponding to candidates, i.e.,  $V(G) = C$ . We construct the set of edges by starting from a complete graph and removing some of them as follows. 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 edge  $\{x, y\}$  from the graph. It is clear that the reduction can be computed in polynomial time and both parameters have the same value. It remains to show its correctness.

First, 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. Indeed, if  $K$  contained two vertices  $x$  and  $y$  that were not connected by an edge, then edge  $\{x, y\}$  had to be removed by some two matched voters  $v$  and  $u$  such that  $x \succ_v y$  and  $y \succ_u x$ . Since both voters belong to subelections  $E'_1$  and  $E'_2$ , we obtain a contradiction that they are isomorphic via the given matchings.

For the other direction, 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  $v$  and  $u$  and an arbitrary pair of candidates  $x, y \in K$ . It follows that  $x \succ_v y$  and  $x \succ_u y$ . Otherwise edge  $\{x, y\}$  would have been removed during the reduction, hence  $K$  would not be a clique. A contradiction.  $\square$

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. Further, a trivial approximation algorithm which returns a constant size solution (hence the approximation ratio is  $O(m)$ ) is also essentially optimal.

**Proposition 4.7.** 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 the problem is in XP for the parameter  $k$ ). Moreover, assuming ETH, there is no  $O^*(m^{o(k)})$ -time algorithm.*

*Proof.* Note that in the input of the problem we have the same number of candidates and the same number of voters in both elections. The algorithm simply guesses  $k$  candidates from one of the input elections and checks if both obtained candidate-subelections restricted to chosen  $k$  candidates are the same. There are  $\binom{m}{k}$  many choices for  $k$  candidates, hence the total running time is bounded by  $O^*(m^k)$ .

Under ETH there is no  $|V|^{o(K)}$ -time algorithm for finding a size- $K$  clique on a graph with vertex set  $V$  (Chen et al., 2006). In our reduction in Theorem 4.6 we have  $m = |V|$  many candidates and required candidate set size is  $k = K$ . Therefore, using an  $O^*(m^{o(k)})$ -time algorithm for MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS we could solve CLIQUE in time  $m^{o(k)} \cdot \text{poly}(m^2) \leq |V|^{o(k)}$ . This contradicts ETH.  $\square$

**Proposition 4.8.** MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS *has a polynomial-time  $t/c$ -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 a  $t^{1-\epsilon}$  factor is NP-hard for every  $\epsilon > 0$ .*

*Proof.* First, note that we use the convention that an approximation ratio  $\alpha$  is at least 1 also for a maximization problem. Therefore, if  $\text{ALG}$  is a value of a solution returned by an algorithm and  $\text{OPT}$  is the value of an optimal solution then  $\frac{\text{OPT}}{\text{ALG}} \leq \alpha$  implies that the solution is an  $\alpha$ -approximation of an optimal one.

Let us describe the algorithm. For any fixed natural number  $c \geq 1$  we check all size- $c$  subsets of candidates as a solution. There are at most  $\binom{m}{c} \leq m^c$  many such subsets (i.e., polynomially many). We evaluate each of them in polynomial time. At least one such trial leads to finding a subset of an optimal solution, so it is a feasible solution of size  $c$ . Therefore the algorithm has approximation ratio at most  $\frac{\text{OPT}}{\text{ALG}} = \frac{t}{c}$ .

It is NP-hard to approximate CLIQUE within factor  $|V|^{1-\epsilon}$  for every  $\epsilon > 0$ , where  $V$  is a given set of vertices (Zuckerman, 2007). In the reduction in Theorem 4.6 the number of candidates  $m$  equals to the number of vertices of the graph, hence  $t \leq m = |V|$ . Therefore, using a  $t^{1-\epsilon}$ -approximation algorithm for MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS, for some  $\epsilon > 0$ , we could approximate CLIQUE within  $|V|^{1-\epsilon}$  factor. This would imply  $\text{P} = \text{NP}$ .  $\square$

As a comment we point that the above running time lower-bound (Proposition 4.7) and hardness of approximation (Proposition 4.8) can be combined under a stronger hypothesis. The work of Chalermsook et al. (2017) give an evidence that for CLIQUE, where  $K$  denotes the size of maximum clique,  $K^{1-\epsilon}$ -approximation is not possible even in time FPT with respect to  $K$  assuming the gap version of ETH, called Gap-ETH. Hence, the same hardness holds for MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS. It means, that the best way to solve both problems, even approximately, is to essentially enumerate all possibilities (Chalermsook et al., 2017).

MAX. COMMON CAND.-SUBELECTION (without any matchings) also is NP-complete, and so are its variants with a candidate matching and with a voter matching. The proofs either follow by applying Proposition 3.1 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 (i.e., the clique size).

**Proposition 4.9.** MAX. COMMON CAND.-SUBELECTION is NP-complete and so are its variants with a given candidate matching and with a given voter matching.

*Proof.* Below we give the reductions for all the three cases, i.e., the case with a given candidate matching, with a given voter matching, and without any matchings.

**Given Candidate Matching.** We give a reduction from MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS. Let  $E_1 = (C, V)$  and  $E_2 = (C, U)$  be our input elections, where  $V = (v_1, \dots, v_n)$  and  $U = (u_1, \dots, u_n)$ , and let  $t$  be the desired size of the isomorphic subelection (since we are in the setting with both matchings, we can assume that both elections are over the same candidate set). We assume that for each  $i \in [n]$ , voter  $v_i$  is matched to  $u_i$ . Let  $m = |C|$  and let  $k = t/n$ . We note that  $k \leq m$ .

Our construction proceeds as follows. First, we form  $m + 1$  sets,  $A, D_1, \dots, D_m$ , each containing  $m + 1$  new candidates. Let  $\mathcal{D} = A \cup D_1 \cup \dots \cup D_m$ . Note that  $|\mathcal{D}| = (m + 1)^2$ . We form elections  $E'_1 = (C \cup \mathcal{D}, V')$  and  $E'_2 = (C \cup \mathcal{D}, U')$ , where  $V' = (v'_1, \dots, v'_n)$  and

$U' = (u'_1, \dots, u'_n)$ . For each  $i \in [n]$ , we set the preference orders of  $v'_i$  and  $u'_i$  as follows (by writing  $[v_i]$  and  $[u_i]$  we mean copying the preference order of  $v_i$  and  $u_i$ , respectively):

$$\begin{aligned} v'_i: D_1 \succ \dots \succ D_{i-1} \succ A \succ D_i \succ \dots \succ D_m \succ [v_i], \\ u'_i: D_1 \succ \dots \succ D_{i-1} \succ A \succ D_i \succ \dots \succ D_m \succ [u_i]. \end{aligned}$$

Finally, we set the desired size of the isomorphic subelections to be  $t' = n \cdot (k + (m + 1)^2) = t + n(m + 1)^2$ .

We claim that  $E'_1$  and  $E'_2$  have isomorphic candidate subelections of size  $t'$  for the given candidate matching if and only if  $E_1$  and  $E_2$  have isomorphic candidate subelections of size  $t$  for given candidate and voter matchings.

Let us assume that  $E'_1$  and  $E'_2$  have the desired candidate subelections,  $E''_1$  and  $E''_2$ . We claim that their isomorphism is witnessed by such a matching that for each  $i$  voter  $v'_i$  is matched to  $u'_i$ . If it were not the case, then to maintain the isomorphism these subelections would have to lose at least  $m - 1$  candidates from  $\mathcal{D}$  (e.g., the candidates from  $A$ ) and their sizes would be, at most,  $n(m + m(m + 1)) = n((m + 1)^2 - 1) < t'$ . Thus the isomorphism of  $E''_1$  and  $E''_2$  is witnessed by the same voter matching as the one required by our input instance. A simple counting argument shows that after dropping candidates from  $\mathcal{D}$  from subelections  $E''_1$  and  $E''_2$ , we obtain elections witnessing that  $(E_1, E_2)$  is a *yes*-instance of MAX. COMMON CAND.-SUBELECTION WITH BOTH MATCHINGS. The reverse direction is immediate.

**Given Voter Matching.** This case follows by Proposition 3.1 and the fact that CAND.-SUBELECTION WITH VOTER MATCHING is NP-complete.

**No Matchings Given.** This case follows by Proposition 3.1 and the fact that CAND.-SUBELECTION ISOMORPHISM problem is NP-complete.  $\square$

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

*Proof.* For the case without any matchings and the case with the voter matching, we use Proposition 3.1 to reduce from the corresponding variant of SUBELECTION ISOMORPHISM. For the variants that include the candidate matching (for which SUBELECTION ISOMORPHISM is in P), we reduce from the corresponding variants of MAX. COMMON CAND.-SUBELECTION. Let  $E_1 = (C, V_1)$  and  $E_2 = (C, V_2)$  be our input elections and let  $t$  be the desired size of their isomorphic candidate subelections (since we are in the “with candidate matching” regime, we take the candidate sets to be equal). Without loss of generality, we can assume that  $|V_1| = |V_2|$ ; our NP-completeness proofs for MAX. COMMON CAND.-SUBELECTION give such instances.

Let  $m = |C|$ ,  $n = |V_1| = |V_2|$ , and let  $D$  be a set of  $(n - 1)m$  dummy candidates. We form elections  $E'_1$  and  $E'_2$  to be identical to  $E_1$  and  $E_2$ , respectively, except that they also include the candidates from  $D$ , who are always ranked on the bottom, in the same order. Hence, the number of candidates in  $E'_1$  and  $E'_2$  equals  $nm$ . We ask if  $E'_1$  and  $E'_2$  have isomorphic subelections of size  $t' = t + n(n - 1)m$ .



If  $E_1$  and  $E_2$  have isomorphic candidate subelections of size  $t$ , then certainly  $E'_1$  and  $E'_2$  have isomorphic subelections of size  $t'$  (it suffices to take the same subelections as for  $E_1$  and  $E_2$  and include the candidates from  $D$ ).

On the other hand, if  $E'_1$  and  $E'_2$  have isomorphic subelections of size  $t'$ , then  $E_1$  and  $E_2$  have size- $t$  isomorphic candidate subelections. Indeed, the subelections of  $E'_1$  and  $E'_2$  must include all the  $n$  voters. Otherwise their sizes would be at most  $(n-1)mn < t+(n-1)mn \leq t'$ . Thus the subelections of  $E'_1$  and  $E'_2$  are candidate subelections. As we can also assume that the subelections of  $E'_1$  and  $E'_2$  include all the candidates from  $D$ , by omitting these candidates we get the desired candidate subelections of  $E_1$  and  $E_2$ .  $\square$

## 5. Experiments

In this section we use the MAX. COMMON VOTER-SUBELECTION problem to analyze similarities 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. Our ILP formulation is available in Appendix A. The source code used for the experiments is available in a GitHub repository<sup>3</sup>. Whenever we discuss the running time of a particular algorithm, we report experiments performed on a single thread on Apple MacBook Air with M1 processor and 8 GB RAM.

We stress that we could have used other problems from the MAX. COMMON SUBELECTION family in this section, but we chose MAX. COMMON VOTER-SUBELECTION because its outcomes are particularly easy to interpret. Specifically, given two elections,  $E_1$  and  $E_2$ , both with the same number of voters, using the MAX. COMMON VOTER-SUBELECTION problem we can compute the largest fraction of votes that we can keep in these elections while ensuring their isomorphism. For example, if we know that it is possible to keep certain 70% of the votes while ensuring the elections' isomorphism, then we have a strong argument that these elections are structurally very similar. On the flip side, if the fraction of the voters in the largest isomorphic subelections is small, then it does not need to mean that the elections are not similar. Indeed, the elections *may* be similar, but their votes may differ slightly, preventing the existence of large isomorphic subelections. For example, consider two elections over a large number of candidates and with a large number of voters, where in the first one all the votes are identical, and in the second one all the votes are identical except that each voter ranks some small number of bottom-ranked candidates differently. We would typically view such elections as very similar, but their largest isomorphic voter subelections contain only a single voter each. While this clearly is a drawback of the isomorphism-based approach, it is the necessary price to pay for the strength of the similarity results. If one insists on identifying such approximate similarity, then one should use the isomorphic swap distance of Faliszewski et al. (2019) or the positionwise distance of Szufa et al. (2020).

Our main finding is that elections with few candidates (say, four) are quite similar to each other irrespective of what model is used to generate them. For larger candidate sets some similarities (in terms of large isomorphic voter subelections) between elections generated from various sources exist, but they are far less frequent.

---

3. <https://github.com/Project-PRAGMA/Subelections>

## 5.1 Statistical Models of Elections

Below we describe several standard models for generating elections. For some of these models it suffices to provide a distribution over the votes, possibly parameterized in some way. Under such models, generating an election boils down to sampling the required number of voters from the given distribution. For other models, we give an explicit procedure for generating an election with a given number of candidates and voters:

**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 & Slinko, 2006). We start with an urn containing exactly one copy of each possible preference order over the given candidate set (we assume that there are  $m$  candidates). 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 putting it back to the urn together with  $\alpha \cdot m!$  duplicates.

**Mallows Model.** The Mallows Model (Mallows, 1957) is parameterized by a value  $\phi \in [0, 1]$  and a central preference order  $u$ , which we choose uniformly at random. The probability of generating a preference order  $v$  is proportional to  $\phi^{\text{swap}(u,v)}$ , where  $\text{swap}(u, v)$  is the minimum number of swaps of adjacent candidates needed to transform  $v$  into  $u$ . In our experiments we do not set the value of  $\phi$  directly, but we use the parameterization by norm- $\phi$  proposed by Boehmer et al. (2021) (strictly speaking, they used parameter  $\text{rel-}\phi$  which is equal to  $0.5 \cdot \text{norm-}\phi$ ). It works as follows: For a given norm- $\phi$  value and a given number  $m$  of candidates in the election to be generated, we choose value  $\phi$  so that for a generated vote  $v$  the expected value of  $\text{swap}(u, v)$  is equal to norm- $\phi$  times half the maximum possible number of swaps between two preference orders (i.e., times  $1/4 \cdot m(m - 1)$ ). 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 for the original Mallows model, and the work of Boehmer et al. (2021) for ways of computing  $\phi$  given norm- $\phi$ .

**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.

Whenever we generate a single-peaked election, we choose the axis by selecting it uniformly at random among all possible ones.

## 5.2 Results and Analysis

We study the following nine models: IC, 1D-Interval, Conitzer model, Walsh model, urn (with  $\alpha \in \{0.1, 0.5\}$ ), Norm-Mallows (with norm- $\phi \in \{1/3, 2/3\}$ ), and identity. We consider elections with 4, 6, 8, and 10 candidates and with 50 voters. For each scenario and each two of the selected models, we have generated 1000 pairs of elections. For each pair of models, we recorded the average fraction of voters in the maximum common voter subelections (expressed as a percentage value), 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 (the 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, up to four or five, it is not as relevant to consider very different election models, but for more candidates using diverse models is justified.

Despite the above, some models remain similar even for 6, 8, and sometimes even 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 6 candidates, but for 8 and 10 candidates it starts to stand out. This is in sync with the maps of elections of Szufa et al. (2020) and Boehmer et al. (2021), where 1D-Interval and Conitzer elections are very close to each other. On these maps, Walsh elections are notably distinct from 1D-Interval and Conitzer ones, but Szufa et al. (2020) and Boehmer et al. (2021) considered elections with 100 candidates and with 10 candidates, so this agrees with our findings.

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 Norm-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 Norm-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 Norm-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 the self-similarity of our models. Intuitively, the larger are these values, the fewer elections of a given

	IC	ID Int.	SP Con.	SP Wal.	Urn-0.1	Urn-0.5	Mal-0.33	Mal-0.67	ID
IC	72 <small>±4</small>	33 <small>±5</small>	41 <small>±4</small>	44 <small>±4</small>	52 <small>±6</small>	31 <small>±7</small>	46 <small>±5</small>	66 <small>±5</small>	11 <small>±2</small>
ID Int.	33 <small>±5</small>	65 <small>±12</small>	67 <small>±10</small>	54 <small>±11</small>	41 <small>±8</small>	47 <small>±10</small>	47 <small>±6</small>	36 <small>±7</small>	43 <small>±13</small>
SP Con.	41 <small>±4</small>	67 <small>±10</small>	83 <small>±5</small>	69 <small>±8</small>	45 <small>±6</small>	41 <small>±8</small>	47 <small>±5</small>	45 <small>±5</small>	29 <small>±5</small>
SP Wal.	44 <small>±4</small>	54 <small>±11</small>	69 <small>±8</small>	82 <small>±5</small>	47 <small>±7</small>	37 <small>±8</small>	55 <small>±6</small>	54 <small>±5</small>	20 <small>±3</small>
Urn-0.1	52 <small>±6</small>	41 <small>±8</small>	45 <small>±6</small>	47 <small>±7</small>	50 <small>±6</small>	40 <small>±9</small>	48 <small>±7</small>	52 <small>±6</small>	26 <small>±8</small>
Urn-0.5	31 <small>±7</small>	47 <small>±10</small>	41 <small>±8</small>	37 <small>±8</small>	40 <small>±9</small>	50 <small>±11</small>	46 <small>±8</small>	34 <small>±8</small>	52 <small>±17</small>
Mal-0.33	46 <small>±5</small>	47 <small>±6</small>	47 <small>±5</small>	55 <small>±6</small>	48 <small>±7</small>	46 <small>±8</small>	75 <small>±6</small>	57 <small>±6</small>	38 <small>±7</small>
Mal-0.67	66 <small>±5</small>	36 <small>±7</small>	45 <small>±5</small>	54 <small>±5</small>	52 <small>±6</small>	34 <small>±8</small>	57 <small>±6</small>	68 <small>±5</small>	16 <small>±4</small>
ID	11 <small>±2</small>	43 <small>±13</small>	29 <small>±5</small>	20 <small>±3</small>	26 <small>±8</small>	52 <small>±17</small>	38 <small>±7</small>	16 <small>±4</small>	100 <small>±0</small>

(a) 4 candidates & 50 voters

	IC	ID Int.	SP Con.	SP Wal.	Urn-0.1	Urn-0.5	Mal-0.33	Mal-0.67	ID
IC	19 <small>±2</small>	9 <small>±2</small>	12 <small>±1</small>	13 <small>±2</small>	11 <small>±2</small>	7 <small>±1</small>	15 <small>±2</small>	18 <small>±2</small>	4 <small>±1</small>
ID Int.	9 <small>±2</small>	50 <small>±12</small>	51 <small>±8</small>	32 <small>±7</small>	22 <small>±6</small>	31 <small>±9</small>	20 <small>±4</small>	11 <small>±2</small>	31 <small>±10</small>
SP Con.	12 <small>±1</small>	51 <small>±8</small>	66 <small>±5</small>	46 <small>±7</small>	20 <small>±4</small>	21 <small>±4</small>	24 <small>±4</small>	15 <small>±2</small>	20 <small>±4</small>
SP Wal.	13 <small>±2</small>	32 <small>±2</small>	46 <small>±7</small>	59 <small>±5</small>	15 <small>±2</small>	12 <small>±3</small>	32 <small>±4</small>	19 <small>±2</small>	9 <small>±2</small>
Urn-0.1	11 <small>±2</small>	22 <small>±6</small>	20 <small>±4</small>	15 <small>±2</small>	20 <small>±4</small>	23 <small>±6</small>	15 <small>±3</small>	11 <small>±2</small>	23 <small>±9</small>
Urn-0.5	7 <small>±1</small>	31 <small>±10</small>	21 <small>±8</small>	12 <small>±4</small>	23 <small>±3</small>	39 <small>±11</small>	13 <small>±8</small>	8 <small>±2</small>	51 <small>±19</small>
Mal-0.33	15 <small>±2</small>	20 <small>±4</small>	24 <small>±4</small>	32 <small>±4</small>	15 <small>±3</small>	13 <small>±3</small>	40 <small>±5</small>	23 <small>±3</small>	12 <small>±3</small>
Mal-0.67	18 <small>±2</small>	11 <small>±2</small>	15 <small>±2</small>	19 <small>±2</small>	11 <small>±3</small>	8 <small>±2</small>	23 <small>±3</small>	22 <small>±2</small>	4 <small>±1</small>
ID	4 <small>±1</small>	31 <small>±10</small>	20 <small>±4</small>	9 <small>±2</small>	23 <small>±9</small>	51 <small>±19</small>	12 <small>±3</small>	4 <small>±1</small>	100 <small>±0</small>

(b) 6 candidates & 50 voters

	IC	ID Int.	SP Con.	SP Wal.	Urn-0.1	Urn-0.5	Mal-0.33	Mal-0.67	ID
IC	5 <small>±1</small>	4 <small>±1</small>	4 <small>±1</small>	5 <small>±1</small>	4 <small>±1</small>	3 <small>±1</small>	5 <small>±1</small>	5 <small>±1</small>	2 <small>±1</small>
ID Int.	4 <small>±1</small>	38 <small>±9</small>	40 <small>±7</small>	15 <small>±3</small>	19 <small>±5</small>	26 <small>±10</small>	8 <small>±2</small>	4 <small>±1</small>	26 <small>±12</small>
SP Con.	4 <small>±1</small>	40 <small>±7</small>	51 <small>±5</small>	22 <small>±4</small>	15 <small>±4</small>	16 <small>±4</small>	10 <small>±2</small>	5 <small>±1</small>	16 <small>±4</small>
SP Wal.	5 <small>±1</small>	15 <small>±3</small>	22 <small>±4</small>	31 <small>±5</small>	6 <small>±1</small>	6 <small>±1</small>	13 <small>±2</small>	6 <small>±1</small>	5 <small>±1</small>
Urn-0.1	4 <small>±1</small>	19 <small>±5</small>	15 <small>±4</small>	6 <small>±1</small>	17 <small>±4</small>	21 <small>±7</small>	5 <small>±1</small>	4 <small>±1</small>	21 <small>±8</small>
Urn-0.5	3 <small>±1</small>	26 <small>±10</small>	16 <small>±4</small>	6 <small>±1</small>	21 <small>±7</small>	41 <small>±11</small>	4 <small>±1</small>	4 <small>±1</small>	49 <small>±19</small>
Mal-0.33	5 <small>±1</small>	8 <small>±2</small>	10 <small>±2</small>	13 <small>±2</small>	5 <small>±1</small>	4 <small>±1</small>	13 <small>±2</small>	7 <small>±1</small>	4 <small>±1</small>
Mal-0.67	5 <small>±1</small>	4 <small>±1</small>	5 <small>±1</small>	6 <small>±1</small>	4 <small>±1</small>	4 <small>±1</small>	7 <small>±1</small>	6 <small>±1</small>	2 <small>±1</small>
ID	2 <small>±1</small>	26 <small>±12</small>	16 <small>±4</small>	5 <small>±1</small>	21 <small>±8</small>	49 <small>±19</small>	4 <small>±1</small>	2 <small>±1</small>	100 <small>±0</small>

(c) 8 candidates & 50 voters

	IC	ID Int.	SP Con.	SP Wal.	Urn-0.1	Urn-0.5	Mal-0.33	Mal-0.67	ID
IC	3 <small>±1</small>	2 <small>±1</small>	3 <small>±1</small>	3 <small>±1</small>	2 <small>±1</small>	2 <small>±0</small>	3 <small>±1</small>	3 <small>±1</small>	2 <small>±0</small>
ID Int.	2 <small>±1</small>	31 <small>±8</small>	31 <small>±6</small>	7 <small>±2</small>	17 <small>±5</small>	21 <small>±8</small>	4 <small>±1</small>	2 <small>±1</small>	21 <small>±9</small>
SP Con.	3 <small>±1</small>	31 <small>±6</small>	38 <small>±5</small>	10 <small>±2</small>	12 <small>±3</small>	13 <small>±3</small>	4 <small>±1</small>	3 <small>±1</small>	13 <small>±3</small>
SP Wal.	3 <small>±1</small>	7 <small>±2</small>	10 <small>±2</small>	12 <small>±2</small>	4 <small>±1</small>	4 <small>±1</small>	5 <small>±1</small>	4 <small>±1</small>	4 <small>±1</small>
Urn-0.1	2 <small>±1</small>	17 <small>±5</small>	12 <small>±3</small>	4 <small>±1</small>	17 <small>±4</small>	21 <small>±7</small>	2 <small>±1</small>	2 <small>±1</small>	21 <small>±8</small>
Urn-0.5	2 <small>±0</small>	21 <small>±8</small>	13 <small>±3</small>	4 <small>±1</small>	21 <small>±7</small>	38 <small>±11</small>	2 <small>±1</small>	2 <small>±0</small>	49 <small>±17</small>
Mal-0.33	3 <small>±1</small>	4 <small>±1</small>	4 <small>±1</small>	5 <small>±1</small>	2 <small>±1</small>	2 <small>±1</small>	5 <small>±1</small>	4 <small>±1</small>	2 <small>±1</small>
Mal-0.67	3 <small>±1</small>	2 <small>±1</small>	3 <small>±1</small>	4 <small>±1</small>	2 <small>±1</small>	2 <small>±0</small>	4 <small>±1</small>	3 <small>±1</small>	2 <small>±0</small>
ID	2 <small>±0</small>	21 <small>±9</small>	13 <small>±3</small>	4 <small>±1</small>	21 <small>±8</small>	49 <small>±17</small>	2 <small>±1</small>	2 <small>±0</small>	100 <small>±0</small>

(d) 10 candidates & 50 voters

Figure 1: The numbers typeset in large font denote the rounded percentage of matched votes for MAX. COMMON VOTER-SUBELECTION. The numbers typeset in small font denote the rounded standard deviation. There are results for elections with 4, 6, 8, and 10 candidates and 50 voters.

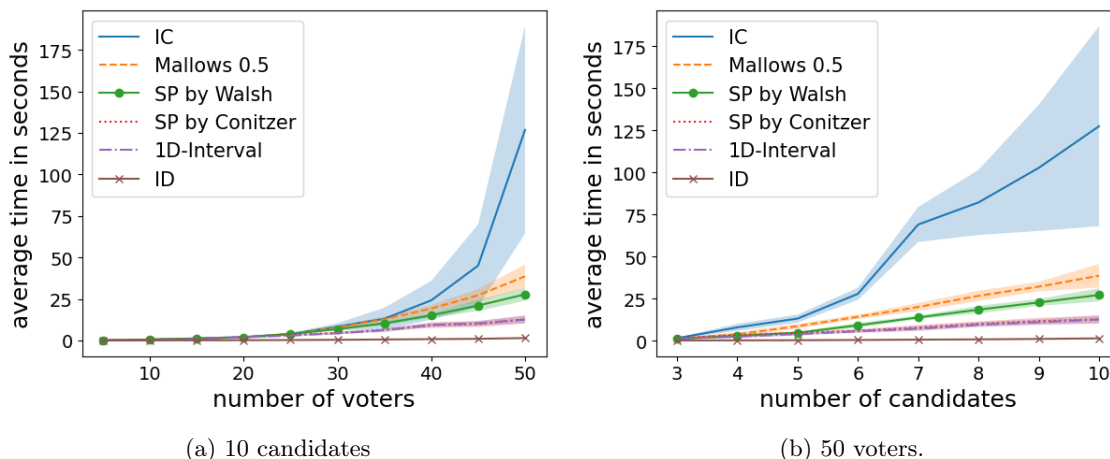


Figure 2: Average time needed to find the maximum common voter subelections with the fixed number of candidates (left), and fixed number of voters (right). The shaded parts depict the standard deviation.

type one needs in an experiment. Single-peaked elections stand out here for all numbers of candidates, whereas 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. We focused on IC, 1D-Interval, Conitzer model, Walsh model, Norm-Mallows model with  $\text{norm-}\phi = 0.5$ , and identity. First, we generated 1000 pairs of elections from each model with 10 candidates and 5, 10, . . . , 45, 50 voters (in each pair both elections are from the same model), 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, . . . , 9, 10 candidates, and, like before, calculated the average time needed to find the maximum common voter subelections.

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 requiring by far the most time. Conitzer and Walsh models are significantly different from each other, even though both generate single-peaked elections. Moreover, the fact that the 1D-Interval and Conitzer models need on average the same amount of time reinforces the arguments about their similarity.

### 5.3 Real-Life Subelections

We also conducted analogous experiments regarding 11 real-life elections from PrefLib (Mattei & Walsh, 2013). Specifically, we analyzed city council elections in Glasgow and Aspen, elections from Dublin North and Meath constituencies (Irish), elections held by non-profit organizations, trade unions, and professional organizations (ERS), data from Tour de France (TDF) (Boehmer & Schaar, 2023), Giro d’Italia (GDI) (Boehmer & Schaar, 2023), speed skating (Boehmer et al., 2021), figure skating,<sup>4</sup> as well as preferences over Sushi (Kamishima,

4. In the figure skating dataset, the votes correspond to how the judges ranked the competitors in a given contest. In races, such as Tour the France, each edition is an election where each vote is a stage of the race and the candidates—the bikers—are ranked according to the place on which they finished the stage.

2003), T-Shirt designs, and costs of living and population in different cities (Caragiannis et al., 2019). We also included impartial culture elections in this experiment, as a reference point. All these elections were also visualized on a map of elections by Boehmer et al. (2021) and we generally used the same preprocessing steps as they did (see the details below).

Except for the Aspen, Sushi, and the T-Shirt preferences, each of the other elections comes from a somewhat larger dataset that includes more than a single election (for example, there were many editions of the Tour de France race, and each corresponds to an election). In each such case, we selected a single election from the respective dataset to focus on.<sup>5</sup> We did so because even elections from the same dataset can be quite different from each other, as witnessed by the results of Boehmer et al. (2021) (on their map, elections from the same source tend to cluster together, but the clusters do have nonnegligible diameters, and some elections are outliers with respect to these clusters).

Further, we need elections where all the voters have total orders over the candidates, but the original data includes a number of ties. We deal with them in the same way as Boehmer et al. (2021), i.e., we perform the following operations:

1. If a given vote includes a tie (except for a tie regarding bottom-ranked candidates only) then we break this tie uniformly at random.
2. If a vote is represented as a top-truncated preference order (i.e., it ranks some number of candidates, who are then followed by a group tied for the bottom positions) then we extend it as follows: (a) We find all the votes whose top part agrees with that of the current vote and which rank at least one further candidate; (b) we select one of them uniformly at random; and (c) we extend the current vote with the candidate that the selected vote ranked at the next position (after the initial part on which the votes agree). This way our vote has one tied-at-the-bottom candidate less. We repeat this process until the vote is complete. If during the process it turns out that there are no votes that agree with the top part of the current one and rank at least one candidate more, then we choose the ordering of the remaining candidates uniformly at random.

By performing these operations, we obtain 11 elections, which we refer to as the *source elections*, where all the voters have total orders over all the available candidates. We summarize these elections (and, in particular, the number of voters that they include) in Table 2.

In the experiment, we consider elections with 4, 6, 8, and 10 candidates, and with 50 voters. We generate these elections by treating the source ones as distributions. Specifically, if we need an election with  $m$  candidates and  $n$  voters, then we first remove from the source election all but  $m$  candidates with the highest Borda scores (i.e., we keep  $m$  candidates that, on average, are ranked highest in this source election), and then we sample with replacement  $n$  votes from the source election. Note that in some source elections—in particular those related to sport events—the number of votes is smaller than 50, which means that elections generated using them certainly include several copies of some of the source votes.

---

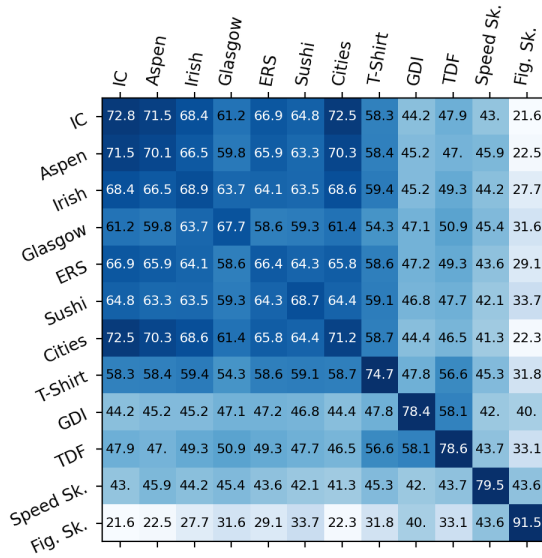
5. In particular, we chose Dublin North election for Irish, Scotland Baillieston Ward for Glasgow, ERS Set 1 for ERS, 2009 Aspen city council election for Aspen, 1998 Euros Men Short Program for figure skating, TDF edition of 1910, GDI edition of 1998, Amateur Competitions Set 1 for speed skating, and cost of living data for the Cities dataset. When choosing these elections, our main guiding principle was that the election includes at least 10 candidates (if possible), and not much more (if possible).

Category	Name	# Votes	# Distinct Votes
Political	Irish	43942	29908
Political	Glasgow	10376	5790
Political	Aspen	2459	2018
Political	ERS	380	336
Sport	Figure Skating	9	9
Sport	Speed Skating	12	12
Sport	TDF	15	15
Sport	GDI	17	17
Survey	T-Shirt	30	30
Survey	Sushi	5000	4926
Survey	Cities	392	392

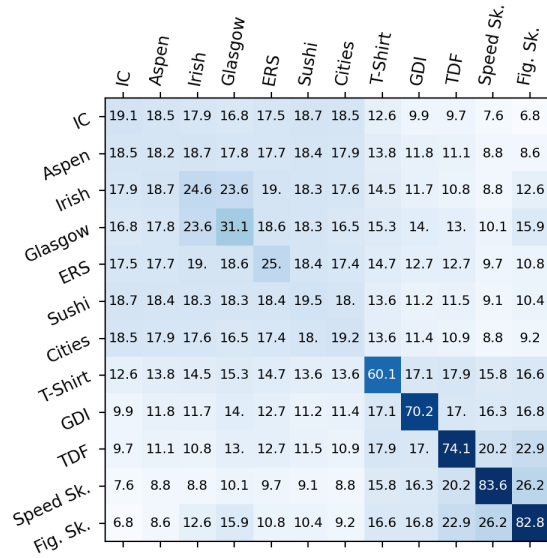
Table 2: Number of votes in the real-life elections used as distributions for sampling.

In Figure 3 we show the results of the experiment on the above-described data (other than the datasets, the experiment is the same as in the preceding section; for example, for each pair of source elections we generated 1000 pairs of elections and computed average number of voters in their maximum voter subelections). For the experiment with only four candidates (the upper left matrix), we see significant similarity between political and survey elections, whereas the sport ones stand out (their self-similarity, i.e., their entries on the diagonal, are high, but not really relevant because they come from source elections with very few votes; this also explains their, much lower but still visible, similarity to each other). Similar phenomena were observed by Boehmer et al. (2021) in their map of real-life elections, albeit, their map considered ten candidates and not four. This difference stems from the fact that our methodology provides very strong similarity results, but is not as good at detecting approximate similarity (whereas the map of elections approach detects approximate similarity well). Thus the interpretation of our results is that for the case of four candidates, there is very strong similarity between political and survey elections, but we cannot say much about the similarity of the sport ones. Interestingly, for the cases of six, eight, and ten candidates, we no longer see the strong similarity of the political and survey elections (which, again, does not mean that they are not similar, but that it is not easy to make them isomorphic by removing just a few voters).

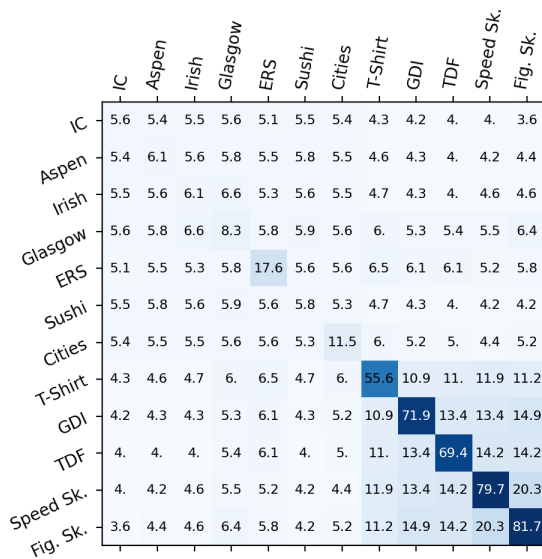
Another thing worth pointing out is the relative similarity between two political elections: Irish and Glasgow. Let us first consider elections with four candidates. When looking for elections that are most similar to those from Glasgow, we find the Irish ones (on average, we can match 67.7% of the votes). When looking for elections that are most similar to the Irish ones, we find Cities (68.6% of the matched votes, on average), IC (68.4%), Aspen (66.5%), ERS (64.1%) and Glasgow (63.7%). For elections with six candidates, both Glasgow and Irish are each other's closest instances and more similar to each other than any other pair (except for the Speed Skating and Figure Skating elections). Nonetheless, this level of similarity is not very high. For the case of eight and ten candidates, the effect is not visible at all. Once again, we stress all the caveats that come with our approach: High similarity values are meaningful and show strong structural similarity between elections, but



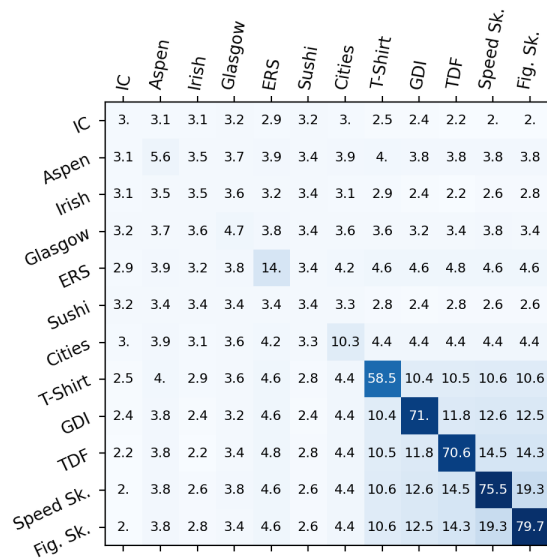
(a) 4 candidates & 50 voters



(b) 6 candidates & 50 voters



(c) 8 candidates & 50 voters



(d) 10 candidates & 50 voters

Figure 3: The numbers denote the rounded percentage of matched votes for MAX. COMMON VOTER-SUBELECTION. There are results for elections with 4, 6, 8, and 10 candidates and 50 voters.



low values do not necessarily carry much information. It would be interesting to verify if similar results hold for other political elections, or if Irish and Glasgow elections are similar by chance.

## 6. Conclusions and Summary

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 MAX. COMMON VOTER-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 establish the parameterized complexity of these problems for the remaining variants.

In Section 5 we have used the MAX. COMMON VOTER-SUBELECTION problem to analyze similarity between both synthetic and real-life elections. We found that for a small number of candidates (such as four) all the synthetic elections that we consider, as well as the political and survey ones, are very similar to each other, but sport-based ones stand out. For larger candidate sets (containing six, eight, or ten candidates) similarities are still visible for some synthetic elections, but essentially disappear for the real-life data. While analyzing these results, one has to keep in mind that isomorphism-based signs of similarity between elections are very strong, but indication of dissimilarity (i.e., the fact that common isomorphic subelections are small) is very weak and, in essence, does not carry information.

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). In particular, this would address the issue with low information content of dissimilarity results. Another possible research direction would be to consider elections with partial preference orders. However, in this case our problems may become notably harder. For example, if we allow top-truncated votes, where each voter ranks some top candidates and leaves the other ones unranked (with the understanding that all unranked candidates are less preferred than the ranked ones), then ELECTION ISOMORPHISM becomes at least as hard as GRAPH ISOMORPHISM: Given a graph  $G$ , we can form a corresponding election where the candidates are the vertices and for each edge  $\{u, v\}$  from the graph, we form two top-truncated votes, one ranking  $u$  above  $v$ , and one ranking  $v$  above  $u$  (and both leaving all the other vertices unranked). Two graphs are isomorphic if and only if their corresponding elections are isomorphic.

## Acknowledgments

An extended abstract of this work appears in AAAI 2022 (Faliszewski, Sornat, & Szufa, 2022). We would like to thank the anonymous AAAI and JAIR reviewers for their valuable comments.

Stanisław Szufa was supported by the National Science Centre, Poland (NCN) grant No 2018/29/N/ST6/01303. Krzysztof Sornat was partially supported by the SNSF Grant 200021\_200731/1. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 101002854).



## References

- Ahuja, R., Magnanti, T., & Orlin, J. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall.
- Arvind, V., Köbler, J., Kuhnert, S., & Vasudev, Y. (2012). Approximate graph isomorphism. In *Proceedings of MFCS-12*, pp. 100–111.
- Babai, L., Dawar, A., Schweitzer, P., & Torán, J. (2015). The graph isomorphism problem (Dagstuhl Seminar 15511). *Dagstuhl Reports*, 5(12), 1–17.
- Ballester, M., & Haeringer, G. (2011). A characterization of the single-peaked domain. *Social Choice and Welfare*, 36(2), 305–322.
- Bartholdi III, J., & 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., Elkind, E., Faliszewski, P., & Szufa, S. (2022). Expected frequency matrices of elections: Computation, geometry, and preference learning. In *Proceedings of NeurIPS-2022*.
- Boehmer, N., Bredereck, R., Faliszewski, P., Niedermeier, R., & Szufa, S. (2021). Putting a compass on the map of elections. In *Proceedings of IJCAI-21*, pp. 59–65.
- Boehmer, N., Cai, J., Faliszewski, P., Fan, A., Janeczko, Ł., Kaczmarczyk, A., & Waś, T. (2023). Properties of position matrices and their elections. In *Proceedings of AAAI-23*, pp. 5507–5514.
- Boehmer, N., Faliszewski, P., Niedermeier, R., Szufa, S., & Waś, T. (2022). Understanding distance measures among elections. In *Proceedings of IJCAI-22*, pp. 102–108.
- Boehmer, N., & Schaar, N. (2023). Collecting, classifying, analyzing, and using real-world ranking data. In *Proceedings of AAMAS-23*, pp. 1706–1715.
- Bokhari, S. (1981). On the mapping problem. *IEEE Transactions on Computers*, 30(3), 207–214.
- Bredereck, R., Chen, J., & Woeginger, G. (2013). A characterization of the single-crossing domain. *Social Choice and Welfare*, 41(4), 989–998.

- Caragiannis, I., Chatzigeorgiou, X., Krimpas, G. A., & Voudouris, A. A. (2019). Optimizing positional scoring rules for rank aggregation. *Artificial Intelligence*, *267*, 58–77.
- Chalermsook, P., Cygan, M., Kortsarz, G., Laekhanukit, B., Manurangsi, P., Nanongkai, D., & Trevisan, L. (2017). From Gap-ETH to FPT-inapproximability: Clique, dominating set, and more. In *Proceedings of FOCS-17*, pp. 743–754.
- Chen, J., Huang, X., Kanj, I. A., & Xia, G. (2006). Strong computational lower 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. A. (1971). The complexity of theorem-proving procedures. In *Proceedings of STOC-71*, pp. 151–158.
- Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2001). *Introduction to Algorithms* (second edition). MIT Press/McGraw Hill.
- Cygan, M., Fomin, F. V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., & Saurabh, S. (2015). *Parameterized Algorithms*. Springer.
- Doignon, J., & Falmagne, J. (1994). A polynomial time algorithm for unidimensional unfolding representations. *Journal of Algorithms*, *16*(2), 218–233.
- Downey, R. G., & 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., & Slinko, A. M. (2012). Clone structures in voters’ preferences. In *Proceedings of ACM EC-12*, pp. 496–513.
- Escoffier, B., Lang, J., & Öztürk, M. (2008). Single-peaked consistency and its complexity. In *Proceedings of ECAI-08*, pp. 366–370.
- Faliszewski, P., Kaczmarczyk, A., Sornat, K., Szufa, S., & Waś, T. (2023). Diversity, agreement, and polarization in elections. In *Proceedings of IJCAI-23*, pp. 2684–2692.
- Faliszewski, P., Skowron, P., Slinko, A., Szufa, S., & Talmon, N. (2019). How similar are two elections?. In *Proceedings of AAI-19*, pp. 1909–1916.
- Faliszewski, P., Sornat, K., & Szufa, S. (2022). The complexity of subelection isomorphism problems. In *Proceedings of AAI-22*, pp. 4991–4998.
- Garey, M., & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.
- Grohe, M., Rattan, G., & Woeginger, G. (2018). Graph similarity and approximate isomorphism. In *Proceedings of MFCS-18*, pp. 20:1–20:16.
- Kamishima, T. (2003). Nantonac collaborative filtering: Recommendation based on order responses. In *Proceedings of KDD-03*, pp. 583–588.
- Kann, V. (1992). On the approximability of the maximum common subgraph problem. In *Proceedings of STACS-92*, pp. 377–388.
- Karp, R. (1972). Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations 1972*, pp. 85–103.

- Lu, T., & 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. *Biometrika*, 44, 114–130.
- Mattei, N., & Walsh, T. (2013). Preflib: A library for preferences <http://www.preflib.org>. In *Proceedings of ADT-13*, pp. 259–270.
- McCabe-Dansted, J., & 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. H. (1994). *Computational Complexity*. Addison-Wesley.
- Raymond, J., & 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., & Talmon, N. (2020). Drawing a map of elections in the space of statistical cultures. In *Proceedings of AAMAS-20*, pp. 1341–1349.
- Vayer, T., Redko, I., Flamary, R., & Courty, N. (2020). CO-optimal transport. In *Proceedings of NeurIPS-20*, pp. 17559–17570.
- Walsh, T. (2015). Generating single peaked votes. *CoRR*, [abs/1503.02766](https://arxiv.org/abs/1503.02766).
- Zuckerman, D. (2007). Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1), 103–128.

## Appendix A. ILP for Maximum Common Voter-Subelection

Since the polynomial-time algorithm for MAX. COMMON VOTER-SUBELECTION is fairly slow, we express the problem as an ILP. We have two types of variables:

1. For each pair of voters  $v \in V$  and  $u \in U$ , we have a binary variable  $N_{v,u}$ . If it is set to 1, then we interpret it as saying that voter  $v$  is included in the subelection of  $E$ , voter  $u$  is included in the subelection of  $F$ , and the two voters are matched. Value 0 means that the preceding statement does not hold.
2. For each pair of candidates  $c \in C$  and  $d \in D$ , we have a binary variable  $M_{c,d}$ . If it is set to 1 then we interpret it as saying that  $c$  is matched to  $d$  in the isomorphic subelections (note that, since we are looking for voter subelections, every candidate from  $C$  has to be matched to some candidate from  $D$ , and the other way round).

To ensure that variables  $N_{v,u}$  and  $M_{c,d}$  describe the respective matchings, we have the following basic constraints:

$$\begin{aligned} \sum_{u \in U} N_{v,u} &\leq 1, \quad \forall v \in V, & \sum_{d \in D} M_{c,d} &= 1, \quad \forall c \in C, \\ \sum_{v \in V} N_{v,u} &\leq 1, \quad \forall u \in U, & \sum_{c \in C} M_{c,d} &= 1, \quad \forall d \in D. \end{aligned}$$

For each pair of voters  $v \in V$ ,  $u \in U$  and each pair of candidates  $c \in C$  and  $d \in D$ , we introduce constant  $w_{v,u,c,d}$  which is set to 1 if  $v$  ranks  $c$  on the same position as  $u$  ranks  $d$ , and which is set to 0 otherwise. We use these constants to ensure that the matchings specified by variables  $N_{v,u}$  and  $M_{c,d}$  indeed describe isomorphic subelections. Specifically, we have the following constraints (let  $m = |C| = |D|$ ):

$$\sum_{c \in C} \sum_{d \in D} w_{v,u,c,d} \cdot M_{c,d} \geq m \cdot N_{v,u}, \quad \forall v \in V, u \in U.$$

For each  $v \in V$  and  $u \in U$ , they ensure that if  $v$  is matched to  $u$  then each candidate  $c$  appears in  $v$  on the same position as the candidate matched to  $c$  appears in  $u$ .

Finally, our objective is to maximize the following sum:

$$\sum_{v \in V, u \in U} N_{v,u}.$$