The Complexity of Election Problems with Group-Separable Preferences

Piotr Faliszewski[1] , **Alexander Karpov**[2,3] and **Svetlana Obraztsova**[4]
[1]AGH University, Krakow, Poland
[2]National Research University Higher School of Economics, Moscow, Russia
[3]Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia
[4]Nanyang Technological University, Singapore
faliszew@agh.edu.pl, akarpov@hse.ru, lana@ntu.edu.sg

Abstract

We analyze the complexity of several NP-hard election-related problems under the assumptions that the voters have group-separable preferences. We show that under this assumption our problems typically remain NP-hard, but we provide more efficient algorithms if additionally the clone decomposition tree is of moderate height.

1 Introduction

We study computational properties of elections for the case where voters' preferences are group-separable [Inada, 1964; 1969]. In this we follow a well-established line of research where the authors assume that the voters' preferences are nicely structured—e.g., are single-peaked [Black, 1958] or single-crossing [Mirrlees, 1971; Roberts, 1977]—and show that various NP-hard election-related problems either become polynomial-time solvable or, less frequently, remain computationally intractable (see, e.g., the survey of Elkind et al. [2016]). So far, group-separable preferences have received limited attention in the literature (especially regarding their computational properties, but see the work of Bredereck et al. [2016]) and we hope that our paper will convince more researchers of their value.

We consider the ordinal model of preferences, that is, we assume that each voter ranks the candidates from the most to the least appealing one. In the general, unrestricted case, each voter is free to report any such preference order, even if—intuitively—it would not correspond to any apparent evaluation of the candidates' merits. By assuming that the preference orders are structured in some way, we require that the voters follow some common, rational criteria. For example, Black [1958] introduced the notion of single-peaked preferences, where all the voters agree on some ordering of the candidates with respect to a given criterion (this ordering is referred to as the *societal axis*; in political elections it may, e.g., correspond to the standard left-to-right spectrum of opinions), each voter has his or her favorite position on the axis (i.e., each voter is free to choose any of the candidates as his or her most preferred one), and given two candidates that lay on the same side of the axis with respect to the voter's favorite candidate, the voter prefers the one closer to the favorite candidate. For example, if a voter ranks a centrist candidate on top, then he or she is forced to prefer a moderate left-wing candidate to an extreme left-wing one.

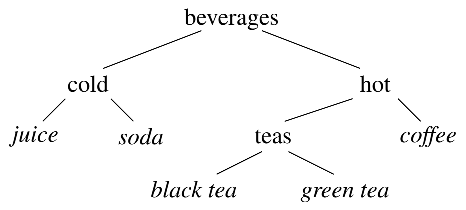

Figure 1: The feature hierarchy (clone decomposition tree) for the beverages example.

Group-separable preferences, introduced by Inada [1964; 1969], also capture a certain rationality assumption, but a completely different one. Formally, we say that the voters have group-separable preferences if each subset A of candidates, $|A| \geq 2$, can be split into two parts, A' and A'', such that each voter prefers all the members of one of these sets to all the members of the other. One interpretation of this is that there is a hierarchy of features that the candidates may have, up to some level of this hierarchy all the candidates from A have the same features, and for the level where they differ, some voters prefer the features of the candidates in A', whereas the others prefer the features of those in A'' (the features do not need to be binary, but this is a natural case).

Example 1. *To illustrate the above intuition, let us consider an example where the voters have preferences over five beverages: coffee, green tea, black tea, juice, and soda. There are three voters with the following preferences (we write $a \succ b$ to indicate that a voter ranks a above b):*

$$v_1\colon \textit{coffee} \succ \textit{black tea} \succ \textit{green tea} \succ \textit{soda} \succ \textit{juice},$$
$$v_2\colon \textit{green tea} \succ \textit{black tea} \succ \textit{coffee} \succ \textit{juice} \succ \textit{soda},$$
$$v_3\colon \textit{soda} \succ \textit{juice} \succ \textit{black tea} \succ \textit{green tea} \succ \textit{coffee}$$

To see that these preferences are group-separable, we make the following observations. First, if we consider the set of all the beverages, then each voter either prefers all the hot drinks to all the cold ones, or the other way round. Second, among the hot drinks, all the voters either prefer coffee to both kinds of tea or the other way round. In other words, all the voters agree that the beverages can be classified as in Figure 1 and form their preferences accordingly (such trees as in Figure 1 *were already discussed by Inada [1964]). In particular, no voter would rank coffee above juice above green tea because it would put a cold drink between two hot ones.*

We note that the tree presented in Figure 1 is, in fact, a *clone decomposition tree* for our preference profile [Elkind *et al.*, 2012]; indeed, all the candidates in a given branch of the tree are *clones* in the sense that all the voters rank them consecutively. While not all decomposition trees correspond to group-separable preferences (see Section 2), for each election there is a polynomial-time computable canonical decomposition tree [Karpov, 2019; Elkind *et al.*, 2012].

Assuming that the voters have single-peaked preferences (or single-crossing ones, or ones that are structured in some other appealing way) has various positive consequences. For example, if the number of voters is odd then there always exists a *Condorcet winner* (i.e., a candidate c that is preferred to every other candidate by a majority of voters). Further, many NP-hard election-related problems become polynomial-time solvable. While it is well-known that group-separable preferences behave equally nicely on the normative front (in particular, they also guarantee the existence of Condorcet winners), we show that their computational properties are more involved. To this end, we consider the problems of winner determination under the Chamberlin–Courant multiwinner voting rule [Chamberlin and Courant, 1983] and Young single-winner voting rule [Young, 1977], as well as two kinds of election control problems. We chose these problems because they all are NP-hard in the general setting, but become polynomial-time solvable if the voters' preferences are single-peaked (or single-crossing). On the high level, our main contribution is that:

> Many (although not all) election-related NP-hard problems remain intractable even if the voters have group-separable preferences. Yet, often these problems can be solved more efficiently (often in polynomial time) if the clone decomposition tree for the given input preference profile has moderate height.

Briefly put, our NP-hardness results rely on the fact that if the clone decomposition tree has caterpillar structure (i.e., it is a path where each node on the path—except for the last one—has an edge to one additional node) then we have enough freedom in forming preference orders to adapt already existing NP-hardness proofs. For our positive results, we either use dynamic programming over the clone decomposition trees (storing an exponential amount of information with respect to the height of the tree) or show reductions to particularly structured integer linear programming (ILP) tasks. We omit some of the proofs due to restricted space.

2 Preliminaries

For an integer t, we write $[t]$ to denote the set $\{1, \dots, t\}$.

An election $E = (C, V)$ consists of a set of candidates $C = \{c_1, \dots, c_m\}$ and a collection $V = (v_1, \dots, v_n)$ of voters. Each voter v_i is associated with preference order $\succ_{v_i}$, which ranks all the candidates from the most to the least desirable one. Given two disjoint subsets of candidates, A and B, we write $v_i \colon A \succ B$ to indicate that v_i prefers each member of A to each member of B. We extend this notation to three or more sets in a natural way, and for singleton sets we often omit braces. For example, given candidates a and b, by $v_i \colon a \succ C \setminus \{a, b\} \succ b$ we mean that voter v_i prefers candidate a to all the other candidates, and prefers all the candidates to candidate b. We omit v_i from this notation if it is clear from the context. For a voter v_i and candidate c, we write $\mathrm{pos}_{v_i}(c)$ to denote the position of c in v_i's preference order (the top-ranked candidate has position 1, the next one has position 2, and so on).

(Weak) Condorcet winners. We say that a candidate c is a *(weak) Condorcet winner* if for every other candidate d a strict majority of the voters (at least half of the voters) ranks c above d. In general, a Condorcet winner may not exist, but if he or she exists, then he or she is unique. A Condorcet winner is also a weak Condorcet winner.

Single-winner voting rules. A single-winner voting rule is a function $\mathcal{R}$ that given an election E outputs a set of tied election winners (in practice, tie-breaking rules are needed, but we disregard this issue[1]). A single-winner voting rule is (weak) Condorcet-consistent if it outputs exactly the Condorcet winner (the weak Condorcet winners) whenever he or she (they) exist. We focus on the following two rules:

1. Under the *Plurality* rule, each candidate gets score equal to the number of voters that rank him or her first, and the candidates with the *highest* score win.
2. Under the *Young* rule [Young, 1977], each candidate gets score equal to the smallest number of voters that need to be removed from the election for this candidate to become a weak Condorcet winner. The candidates with the *lowest* score win.

Naturally, the Young rule is (weak) Condorcet-consistent, whereas the Plurality rule is not.

Example 2. *Consider the election from Example 1. The Plurality rule outputs three tied candidates, coffee, green tea, soda, whereas the Young rule outputs a unique winner, black tea (who is the Condorcet winner).*

Multiwinner rules. Multiwinner voting rules are defined analogously to the single-winner ones, except that their input additionally includes the desired committee size k, and they output a family of committees—i.e., size-k subsets of candidates—that tie as election winners. We are interested in the Chamberlin–Courant (CC) multiwinner rule [Chamberlin and Courant, 1983], which is defined as follows: Let β be the Borda dissatisfaction function, defined for each positive integer i as $\beta(i) = i - 1$. Further, consider an election $E = (C, V)$, committee size k, and some committee $W \subseteq C$. The CC-dissatisfaction score of W is:

$$\mathrm{dissat}_E^{\mathrm{cc}}(W) = \textstyle\sum_{v \in V} \min_{c \in W} \big(\beta(\mathrm{pos}_v(c))\big).$$

Intuitively, each voter v contributes dissatisfaction $\beta(i)$ to the committee, where i is the position of the committee member that v ranks highest; this candidate is referred to as the *representative* of v. The CC rule outputs the size-k committees

[1]Occasionally, tie-breaking may affect the complexity of election problems [Obraztsova and Elkind, 2011; Obraztsova *et al.*, 2011].

with the lowest CC-dissatisfaction score. There are a number of variants of the CC rule which use other functions in place of β (or focus on the approval setting [Procaccia *et al.*, 2008]), but for the sake of brevity we focus on the classic, Borda-based variant.

Example 3. *In the election from Example 1, there are three CC winning committees of size $k = 2$. Each of them contains soda and one of the hot drinks. Their CC-dissatisfaction score is two (e.g., the committee $\{black\ tea, soda\}$ receives one point from each of the first two voters and zero points from the last one).*

Computational problems. We are interested in the following computational problems. In each of the problems below we are given an election $E = (C, V)$, a (preferred) candidate p (except for the CC-SCORE problem, where we are given committee size k instead), and a nonnegative integer t (except for the YOUNG-WINNER problem):

1. In the YOUNG-WINNER problem we ask if p is one of the Young winners of election E. In the YOUNG-SCORE problem we ask if p's Young score is at most t. Both problems are NP-hard [Brandt *et al.*, 2015; Rothe *et al.*, 2003] (indeed, the former problem is Θ_2^p-complete and the latter is NP-complete), but both are polynomial-time solvable for single-peaked [Brandt *et al.*, 2015] and single-crossing [Magiera and Faliszewski, 2017] preferences.
2. In the CC-SCORE problem we ask if there is a size-k committee whose CC-dissatisfaction score is at most t. The problem is NP-complete [Procaccia *et al.*, 2008; Lu and Boutilier, 2011] in general, but becomes polynomial-time solvable under both single-peaked [Betzler *et al.*, 2013] and single-crossing [Skowron *et al.*, 2015b] preferences.
3. In the PLURALITY-CCAC problem[2] we are additionally given a set A of spoiler candidates (the voters' preferences include members of A), whereas in the CONDORCET-CCAV problem[3] we additionally get a collection W of spoiler voters. In the former problem we ask if there is a set $A' \subseteq A$ of at most t spoiler candidates such that p is a Plurality winner of election $(C \cup A', V)$, whereas in the latter we ask if there is a collection W' of at most t spoiler voters such that p is the Condorcet winner of the election $(C, V + W')$. Both problems were shown to be NP-complete by Bartholdi et al. [1992], whereas Faliszewski et al. [2011] and Brandt et al. [2015] showed polynomial-time algorithms for the single-peaked cases, and Magiera and Faliszewski [2017] did the same for single-crossing ones.

We also study the PLURALITY-CCDC and CONDORCET-CCDV problems (constructive control by deleting candidates or voters, respectively). They are defined analogously to the CCAC/CCAV variants, but instead of adding spoiler candidates/voters, we ask if it is possible to make the preferred candidate a winner by deleting at most t candidates or voters, respectively. The difference between CONDORCET-CCDV and YOUNG SCORE is that in the former we ask about a Condorcet winner and in the latter about a weak one.

[2]CCAC stands for constructive control by adding candidates.

[3]CCAV stands for constructive control by adding voters.

3 Group-Separability and Clone Structures

The notion of group-separable preferences is due to Inada [1964; 1969]. We say that an election E is *group separable* (or, that the voters' preferences are group-separable) if each subset $A \subseteq C$ of candidates, $|A| \geq 2$, can be partitioned into two nonempty subsets, A' and A'', such for each voter $v \in V$ it holds that either $v\colon A' \succ A''$ or $v\colon A'' \succ A'$. It turns out that it is particularly convenient to look at group-separable elections through the lenses of clone analysis.

We say that a subset B of candidates is a *clone set* in election E if all the voters rank the members of B consecutively. We write $\mathcal{C}(E)$ to denote the family of all clone sets of election E; we also refer to $\mathcal{C}(E)$ as the *clone structure* of E.

As discussed by Elkind et al. [2012], clone structures are hierarchical and are conveniently described using PQ-trees of Booth and Lueker [1976]: A PQ-tree over candidate set C is an ordered tree, where each leaf is labeled with a unique candidate and each internal node is either of type P or of type Q (for the sake of brevity, we will often treat candidates as if they were the leaves themselves). By reading the candidates associated with the leaves from left to right, we obtain a preference order referred to as the *frontier* of this tree. A preference order $\succ$ is consistent with a given PQ-tree if it can be obtained as its frontier by performing the following types of operations: (a) if a node is of type P, then its children can be permuted arbitrarily, and (b) if a node is of type Q, then the order of its children can be reversed.

If every preference order in an election $E = (C, V)$ is consistent with a given PQ-tree $\mathcal{T}$ (over the same candidate set), then we say that E is consistent with $\mathcal{T}$. We say that $\mathcal{T}$ is a *clone decomposition tree* of E if E is consistent with $\mathcal{T}$ and for each set A of candidates, A is a clone set in E if and only if either (a) there is a subtree of $\mathcal{T}$ whose leaves are exactly the candidates from A, or (b) there is a sequence of subtrees of $\mathcal{T}$, whose leaves are consecutive children of the same Q-node, such that A is the union of the sets of leaves of these trees. Elkind et al. [2012] argued that for every election there is a clone decomposition tree and that it is, in essence, unique (up to permuting the order of children in nodes of type P and reversing the order of children of nodes of type Q). Further, this tree is easily computable in polynomial time. Karpov [2019] has shown that if an election is group-separable, then all the nodes in its clone decomposition tree are of type Q.

Example 4. *Let $E = (C, V)$ be the election from Example 1. In addition to all the singleton sets and the set of all candidates, the clone structure of E consists of sets $\{juice, soda\}$, $\{black\ tea, green\ tea\}$, and $\{black\ tea, green\ tea, coffee\}$. All the preference orders in this election are consistent with the PQ-tree from Figure 1 (all nodes are of type Q, say). Indeed, this tree is a clone decomposition tree for E.*

Let $\mathcal{T}$ be some PQ-tree over candidate set C. For a node T of $\mathcal{T}$, we write $C(T)$ to denote the set of candidates that appear as the leaves of the tree rooted at T. For example, for the tree from Figure 1 we have $C(\text{cold}) = \{juice, soda\}$.

4 Hardness Results and Caterpillar Trees

In this section we argue that nearly all our problems remain NP-hard even if the voters' preferences are group-separable (the YOUNG WINNER problem is the only exception—under group-separable preferences weak Condorcet winners always exist and it suffices to check if the given candidate p is one of them). We obtain all our NP-hardness proofs using the same trick, i.e., by noting that we can form a caterpillar PQ-tree (where all the nodes are type-Q) and ensure that all our preference orders are consistent with it. Caterpillar trees give us enough flexibility to either form simple NP-hardness proofs or adapt those already available in the literature.

A caterpillar tree consists of a core path and each node either lays on the path or is directly connected to a node on the path. In our case we will be interested in binary caterpillar PQ-trees (we let all nodes be of type Q). Let $C = \{c_1, \dots, c_m\}$ be a set of candidates. We write $\mathcal{CP}(c_1, \dots, c_m)$ to mean the following caterpillar tree: There are internal nodes $P_1, \dots, P_{m-1}$, where P_1 is the root of the tree and for each $i \in [m-2]$, P_i has one leaf child c_i and one internal-node child P_{i+1}. Further, P_{m-1} has two leaf children, c_{m-1} and c_m.[4]

Observation 1. *Consider a tree $\mathcal{T} = \mathcal{CP}(c_1, \dots, c_m)$. Let A and B be two disjoint subsets of C such that $C = A \cup B$ (i.e., A and B form a partition of C). Then a preference order v of the form $v\colon A \succ B$, where all the members of A are listed in the order of increasing indices and all the members of B are listed in the order of decreasing indices, is consistent with $\mathcal{T}$.*

Using Observation 1, we can show our first result (similar proofs work for CONDORCET-CCAV/CCDV).

Proposition 1. YOUNG-SCORE *is* NP*-hard even for group-separable elections.*

Proof. We will give a reduction from the following classic variant of the X3C problem. The input instance I consists of a set $X = \{x_1, \dots, x_{3k}\}$ and a family $\mathcal{S} = \{S_1, \dots, S_m\}$ of size-3 subsets of X (each element of X belongs to exactly three sets in $\mathcal{S}$). The question is if it is possible to choose k sets whose union is X. We assume that $k \geq 3$. We form an instance of YOUNG-WINNER with candidate set $C = \{x_1, \dots, x_{3k}, p\}$ and the following voters, whose preference orders are consistent with $\mathcal{CP}(x_1, \dots, x_{3k}, p)$ and, thus, are group-separable:

1. For each set $S_i \in \mathcal{S}$, there is a voter with preference order of the form $X - S_i \succ p \succ S_i$ (in the language of Observation 1, we have $A = X - S_i$ and $B = \{p\} \cup S_i$).
2. We have $m - k - 4$ voters with preference order of the form $p \succ x_{3k} \succ \cdots \succ x_1$.

We ask if p's Young score is at most k, i.e., we ask if it is possible to ensure that p is a weak Condorcet winner by deleting at most k voters.

Prior to deleting any voters, for each candidate x_i there are exactly $(m-k-4)+3 = m-k-1$ voters who prefer p to x_i and $m-3$ voters who prefer x_i to p. Since $(m-3)-(m-k-1) = k-2 > 0$, p is not a weak Condorcet winner.

[4]Note that the core path in this tree is, e.g., $P_1, \dots, P_{m-1}, c_m$.

However, if there is a family of k sets from $\mathcal{S}$ whose union is X, then by deleting the corresponding voters we remove $k-1$ voters who prefer x_i to p and one voter who prefers p to x_i. Then we have $(m-3-(k-1))-(m-k-1-1) = 0$, so p and x_i tie in their head-to-head contest. As this reasoning applies to all members of X, we have that p becomes a weak Condorcet winner. Thus, if I is a yes-instance then we also output a yes-instance.

For the other direction, we assume that it is possible to delete at most k voters so that p becomes a weak Condorcet winner. It does not make sense to delete voters who prefer p to all the other candidates, so we assume that we delete voters from the first group only. By a reasoning analogous to that in the previous paragraph, for each element x_i we have to delete at least $k-1$ voters who prefer x_i to p and at most one who prefers p to x_i. This is possible only if I is a yes-instance. $\square$

Corollary 1. CONDORCET-CCAV *and* CONDORCET-CCDV *are* NP*-hard even for group-separable elections.*

For the case of CC-SCORE, we observe that its NP-hardness proof of Skowron et al. [Skowron *et al.*, 2015a, Theorem 3] has the following properties. The set C of candidates is partitioned into two subsets, $A = \{a_1, \dots, a_n\}$ and $D = \{d_1, \dots, d_m\}$, where D is a set of dummy candidates. Further, the set D is partitioned into groups $D_1, D_2, \dots$ (where each D_i contains consecutive members of D). Each voter has a preference order of the form $a_i \succ a_j \succ D_x \succ C \setminus (D_x \cup \{a_i, a_j\})$, where $i < j$ and x is some integer (and the orders of candidates within D_x and $C \setminus (D_x \cup \{a_i, a_j\})$ are irrelevant). Such preference orders are consistent with caterpillar trees $\mathcal{CP}(a_1, \dots, a_n, d_1, \dots, d_m)$ and, thus, the proof applies to group-separable preferences. We similarly adapt the NP-hardness proofs for PLURALITY-CCAC/CCDC [Bartholdi *et al.*, 1992; Hemaspaandra *et al.*, 2007].

Corollary 2. CC-SCORE*,* PLURALITY-CCAC*, and* PLURALITY-CCDC *are* NP*-hard even for group-separable elections.*

Our NP-hardness proofs use caterpillar trees, which have the largest height possible. In the following sections we show algorithms for the case of trees with more moderate height.

5 Algorithm for the CC Rule

In this section we focus on the CC-SCORE problem, which asks if there is a committee with at least a given CC score.

Theorem 1. *There is an algorithm for* CC-SCORE *that given committee size k and an election E with m candidates and n voters, where the voters' preferences are group-separable and yield a clone decomposition tree of height h, runs in time $O(2^h) \cdot \mathrm{poly}(n, m, k)$.*

While, in general, our algorithm runs in exponential time, we note that if the clone decomposition tree has height $h = O(\log m)$ (which happens, e.g., if it is a balanced binary tree) then the algorithm runs in polynomial time. Our algorithm is based on dynamic programming over the clone decomposition tree of the input election; we describe it throughout the rest of this section. We first note that it suffices to focus on CC committees of a particular form.

Lemma 1. *Let $E = (C, V)$ be a group-separable election and let $\mathcal{T}$ be its clone decomposition tree. Let $S \subseteq C$ be some committee. There exists a committee $S' \subseteq C$, $|S'| \le |S|$, such that $\mathrm{dissat}^{\mathrm{cc}}_E(S') \le \mathrm{dissat}^{\mathrm{cc}}_E(S)$ and for each internal node T of $\mathcal{T}$, whose children are $T_1, \dots, T_r$ (and appear in this order in the tree), exactly one of the following holds:*

1. *at most one of the sets $C(T_1), \dots, C(T_r)$ has a nonempty intersection with S', or*
2. *exactly $C(T_1)$ and $C(T_r)$ have nonempty intersections with S'.*

Proof. If committee S does not satisfy our condition then let T be some node of $\mathcal{T}$ where this condition is violated, and let $T_1, \dots, T_r$ be the children of T (in the order in which they appear in the tree). Let c_1 be some arbitrary member of $C(T_1)$ and let c_r be some arbitrary member of $C(T_r)$. Since the condition is violated, it must be that $S \cap \bigcup_{i=1}^{r} C(T_i)$ contains at least two elements (because at least two sets among $C(T_1), \dots, C(T_r)$ have nonempty intersection with S). We form $S'' = (S \setminus \bigcup_{i=1}^{r} C(T_i)) \cup \{c_1, c_r\}$ and claim that S'' satisfies our condition at node T, contains at most $|S|$ candidates, and does not have higher dissatisfaction than S. Indeed, $|S''| \le |S|$ and exactly $C(T_1)$ and $C(T_r)$ have nonempty intersections with S''. To see why $\mathrm{dissat}^{\mathrm{cc}}_E(S'') \le \mathrm{dissat}^{\mathrm{cc}}_E(S)$, we note that for every voter v in E, his or her preference order either satisfies $v\colon C(T_1) \succ C(T_2) \succ \cdots \succ C(T_r)$, or satisfies $v\colon C(T_r) \succ \cdots \succ C(T_2) \succ C(T_1)$ (this follows because for group-separable preferences the clone decomposition trees contain Q-nodes only). In either case, if the highest-ranked member of S belonged to $\bigcup_{i=1}^{r} C(T_i)$, then the highest ranked member of S'' is either ranked at the same position or higher. Further, we see that for every node where S did not violate the conditions, committee S'' also does not violate them. By repeating the above-described process for all the nodes where the lemma condition is violated we obtain the desired committee. □

We will refer to the committees of the form described in Lemma 1 as *normal*. Our algorithm will compute a lowest-dissatisfaction normal committee of up to a given size k, and then fill it in up to size k with arbitrary candidates. By Lemma 1, this will be a winning committee for our election. To this end, we will need a certain compact way of representing key properties of normal committees.

Let $E = (C, V)$ be some group-separable election and let $\mathcal{T}$ be its clone decomposition tree, whose root is node P_0. Further, let S be a normal committee for E (and $\mathcal{T}$) and let $P = (P_0, ..., P_q)$ be a path leading from the root of $\mathcal{T}$ down to some node P_q (this node may either be an internal node or a leaf). We say that this path is *relevant* if $C(P_q) \cap S \neq \emptyset$ (i.e., if there is some member of S that is reachable from the root of $\mathcal{T}$ by a path that extends P). If P is relevant, then for each P_i, $i \in [q]$, we define $\mathrm{type}(P_i)$ as follows (let $Q_1, \dots, Q_r$ be the children of node P_{i-1}):

1. $\mathrm{type}(P_i) = 1$ if for each Q_j, $j \in [r]$, it holds that $C(Q_j) \cap S \neq \emptyset$ if and only if $Q_j = P_i$ (in other words, P_i is the unique child of P_{i-1} that has leaves in S);
2. $\mathrm{type}(P_i) = 2$ if $r \ge 2$, $C(Q_1) \cap S \neq \emptyset$, and $C(Q_r) \cap S \neq \emptyset$ (i.e., exactly the two extreme children of P_{i-1} have leaves in S; by definition, P_i is one of these children).

Since S is a normal committee, the above two types are the only ones possible. We define the type vector of the path P to be $\mathrm{type}(P) = (\mathrm{type}(P_1), \dots, \mathrm{type}(P_q))$. As we will next show, given P and $\mathrm{type}(P)$ it is possible to compute those voters in V whose representatives (with respect to S) are in $C(P_q)$. We refer to this collection of voters as $V(P, \mathrm{type}(P))$. (A crucial observation here is that this collection of voters does not depend on the actual contents of S, provided that they are consistent with $\mathrm{type}(P)$.)

Lemma 2. *Let $E = (C, V)$ be an election and let $\mathcal{T}$ be its clone decomposition tree. Let S be some normal committee for E and let $P = (P_0, \dots, P_q)$ be some relevant path in $\mathcal{T}$. There is an algorithm that given E, $\mathcal{T}$, P, and* $\mathrm{type}(P)$ *computes $V(P, \mathrm{type}(P))$.*

Proof. Our algorithm proceeds by first setting V' to be the collection of all the voters (i.e., $V' := V$) and then removing voters from V' until we are left with the desired collection. Specifically, we consider nodes $P_1, \dots, P_q$ in order and, while considering node P_i, we proceed as follows (let $Q_1, \dots, Q_r$ be the children of P_{i-1}):

1. If $\mathrm{type}(P_i) = 2$ then we consider two cases. If $P_i = Q_1$ then we remove from V' all the voters v for whom it holds that $v\colon C(Q_r) \succ C(Q_1)$; if $P_i = Q_r$ then we remove from V' all the voters v for whom it holds that $v\colon C(Q_1) \succ C(Q_1)$ (since $\mathcal{T}$ is a clone decomposition tree for a group-separable profile, we know that for each voter one of the above conditions holds).
2. If $\mathrm{type}(P_i) = 1$ then we leave V' unchanged.

A simple induction shows that for each $i \in [q]$, after considering node P_i, V' contains exactly those voters, whose representatives belong to $C(P_i)$. Thus, in the end, $V' = V(P, \mathrm{type}(P))$. It is also easy to verify that the algorithm runs in polynomial time. □

We are now ready to present our main algorithm. The input consists of a group-separable election $E = (C, V)$ and committee size k. Additionally, we also compute E's clone decomposition tree $\mathcal{T}$; this can be done in polynomial time [Elkind *et al.*, 2012]. The algorithm is based on dynamic programming. For each number $k' \in [k]$ of candidates, each path $P = (P_0, \dots, P_q)$ leading from the root of $\mathcal{T}$ to one of its internal nodes or leaves, and each vector $T = (t_1, \dots, t_q) \in \{1, 2\}^q$—which specifies the intended type of P—we define function $f(k', P, T)$ to be the lowest dissatisfaction achievable by a committee $S \subseteq C(P_q)$, $|S| \le k'$, among voters $V(P, T)$. We note that for path $P_{\mathrm{root}} = (P_0)$ and empty vector $T_{\mathrm{root}} = ()$, we have that $f(k, P_{\mathrm{root}}, T_{\mathrm{root}})$ gives exactly the dissatisfaction of a winning size-k committee for election E. It remains to show how to compute f efficiently.

Consider some integer $k' \in [k]$, a path $P = (P_0, \dots, P_q)$, and vector $T = (t_1, \dots, t_q) \in \{1, 2\}^q$. If P_q is a leaf (and c is the candidate associated with this leaf), then $f(k', P, Q) = \mathrm{dissat}^{\mathrm{cc}}_{(C, V(P,T))}(\{c\})$. This follows because, by definition of f, we can only consider committees that are nonempty subsets of $\{c\}$, so there is just one choice. If P_q is not a leaf, then let $Q_1, \dots, Q_r$ be its children within $\mathcal{T}$. For $j \in [r]$, we write $P|Q_j$ to denote $(P_0, \dots, P_q, Q_j)$ and for $t \in \{1,2\}$ we write $T|t$ to denote $(t_1, \dots, t_q, t)$. We express f recursively as $f(k', P, Q) = \max(A, B)$, where:

$$A = \max_{j \in [r]} \quad f(k', P|Q_r, T|1), \text{ and}$$
$$B = \max_{\substack{k'_1, k'_r \in [k'] \\ k'_1 + k'_r = k'}} \quad f(k'_1, P|Q_1, T|2) + f(k'_r, P|Q_r, T|2).$$

Intuitively, value A corresponds to choosing committee members from a tree rooted at a single child of P_q, whereas B corresponds to choosing some committee members from $C(Q_1)$ and some from $C(Q_r)$ (by Lemma 1 there is no need to consider other possibilities).

Using the above recursion and standard dynamic programming, it is possible to compute $f(k, P_{\mathrm{root}}, T_{\mathrm{root}})$ in time $O(2^h) \cdot \mathrm{poly}(|C|, |V|, k)$, where h is the height of $\mathcal{T}$. The $O(2^h)$ factor comes from the fact that for each node of $\mathcal{T}$ we need to consider at most 2^h possible types of paths that lead to it. As argued above, $f(k, P_{\mathrm{root}}, T_{\mathrm{root}})$ is the lowest dissatisfaction achievable by a committee of size (at most) k in our election and computing it suffices to decide CC-SCORE. Via dynamic programming tricks, we can also compute a committee that achieves this dissatisfaction.

6 Plurality Control

For the case of PLURALITY-CCAC and PLURALITY-CCDC we obtain a result analogous to that for CC-SCORE.

Theorem 2. *There are algorithms for the* PLURALITY-CCAC *and* PLURALITY-CCDC *problems that for group-separable elections run in time $O(4^h) \cdot \mathrm{poly}(n, m)$, where h is the height of the clone decomposition tree for the input election, n is the number of its voters, and m is the number of its candidates.*

7 Young Rule and Condorcet Control

We will now present our algorithm for computing the Young score of a given candidate. Unfortunately the algorithm will not work in polynomial time even for elections with decomposition trees of height $O(\log m)$. Still, the running time will be subexponential, giving a strong argument that in this case the problem is not NP-hard.

Theorem 3. *There is an algorithm that given an election E with m candidates and n voters, where the voters' preferences are group-separable and yield clone decomposition tree of height h, computes the Young score of a given candidate in time $h^{O(h^2)} \cdot \mathrm{poly}(n, m)$.*

Let $E = (C, V)$ be our input election and let p be the candidate in whose Young score we are interested. Let $\mathcal{T}$ be the clone decomposition tree for E, and let $P = (P_0, \dots, P_h)$ be the path leading from the root of $\mathcal{T}$ to a leaf node corresponding to p. Our algorithm forms an integer linear program (ILP) and then solves it using the algorithm of Eisenbrand and Weismantel [2018] (see also the overview of Gavenciak et al. [2018] for a discussion of modern ILP algorithms). For each voter $v \in V$, we have a binary variable x_v which takes value 1 if the voter stays in the election (i.e., is not deleted), and which takes value 0 otherwise. To form constraints, we consider each node P_i, $i \in [h-1] \cup \{0\}$, separately, and for each we introduce two constraints. Let us fix some node P_i and let $Q_1, \dots, Q_r$ be its children, such that $Q_j = P_{i+1}$. Let X be the collection of voters v such that $v\colon C(Q_1) \succ \dots \succ C(Q_r)$ (we refer to them as X-voters), and let Y be the set of voters for whom it holds that $v\colon C(Q_r) \succ \dots \succ C(Q_1)$ (we refer to them as Y-voters; by the properties of group-separable preferences, we know that each voter is either of type X or of type Y). We introduce the constraints as follows:

1. If $j > 1$ then for $p \in C(P_j)$ to be a weak Condorcet winner, there must be at least as many X-voters as Y-voters (or some member of $C(Q_1)$ would be strictly preferred to p). We form constraint $\sum_{v \in X} x_v \geq \sum_{v \in Y} x_v$.
2. If $j < r$ then, by an analogous reasoning as above, we require that there are at least as many Y-voters as X-voters and form constraint $\sum_{v \in Y} x_v \geq \sum_{v \in X} x_v$.

One can verify that if all these constraints are satisfied, then p is a weak Condorcet winner. We set the goal in our ILP to be maximizing $\sum_{v \in V} x_v$ (so that we delete as few voters as possible). The score of p is $|V| - \sum_{v \in V} x_v$.

For the running time, we have $|V|$ variables and $O(h)$ constraints. As all the coefficients in our constraints are either 1 or -1, the algorithm of Eisenbrand and Weismantel [2018] runs in time $|V| \cdot h^{O(h^2)}$, which suffices for us. Analogous algorithms also work for Condorcet-CCAV/CCDV.

Corollary 3. *There are algorithms that solve* CONDORCET-CCAV/CCDV *for group-separable elections in time $h^{O(h^2)} \cdot \mathrm{poly}(m, n)$, where h is the height of the clone decomposition tree for the input election, m is the number of candidates, and n is the number of voters.*

8 Conclusions

So far, most algorithmic studies of restricted domains focused on positive results and provided efficient algorithms. This is the case for the single-peaked [Faliszewski *et al.*, 2011; Conitzer, 2009] and single-crossing [Skowron *et al.*, 2015b; Magiera and Faliszewski, 2017] domains, as well as for many of their variants [Peters and Elkind, 2016; Yu *et al.*, 2013]. Indeed, there even are domains—such as SPOC [Peters and Lackner, 2017]—that lack good normative properties, but are useful algorithmically. We have shown that the group-separable domain is somewhat different. It is known to have good normative properties, but to obtain efficient algorithms, one has to look deeper into its structure.

Acknowledgements

This work was supported by the MOE (Singapore) under Tier-1 grant 2018-T1-001-118 and by the NTU SUG M4081985 grant. Alexander Karpov was Supported by the Basic Research Program of the National Research University Higher School of Economics.

References

[Bartholdi *et al.*, 1992] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[Betzler *et al.*, 2013] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *Journal of Art. Int. Research*, 47:475–519, 2013.

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

[Booth and Lueker, 1976] K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976.

[Brandt *et al.*, 2015] F. Brandt, M. Brill, E. Hemaspaandra, and L. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. *Journal of Art. Int. Research*, 53:439–496, 2015.

[Bredereck *et al.*, 2016] R. Bredereck, J. Chen, and G. Woeginger. Are there any nicely structured preference profiles nearby? *Math. Social Sciences*, 79:61–73, 2016.

[Chamberlin and Courant, 1983] B. Chamberlin and P. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, 77(3):718–733, 1983.

[Conitzer, 2009] V. Conitzer. Eliciting single-peaked preferences using comparison queries. *Journal of Art. Int. Research*, 35:161–191, 2009.

[Eisenbrand and Weismantel, 2018] F. Eisenbrand and R. Weismantel. Proximity results and faster algorithms for integer programming using the Steinitz lemma. In *Proceedings of SODA-18*, pages 808–816, 2018.

[Elkind *et al.*, 2012] E. Elkind, P. Faliszewski, and A. Slinko. Clone structures in voters' preferences. In *Proceedings of EC-12*, pages 496–513, June 2012.

[Elkind *et al.*, 2016] E. Elkind, M. Lackner, and D. Peters. Preference restrictions in computational social choice: Recent progress. In *Proceedings of IJCAI-2016*, pages 4062–4065, 2016.

[Faliszewski *et al.*, 2011] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, 209(2):89–107, 2011.

[Gavenciak *et al.*, 2018] T. Gavenciak, D. Knop, and M. Koutecký. Integer programming in parameterized complexity: Three miniatures. In *Proceedings of IPEC-18*, pages 21:1–21:16, 2018.

[Hemaspaandra *et al.*, 2007] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[Inada, 1964] K. Inada. A note on the simple majority decision rule. *Econometrica*, 32(32):525–531, 1964.

[Inada, 1969] K. Inada. The simple majority decision rule. *Econometrica*, 37(3):490–506, 1969.

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

[Lu and Boutilier, 2011] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of IJCAI-11*, pages 280–286, 2011.

[Magiera and Faliszewski, 2017] K. Magiera and P. Faliszewski. How hard is control in single-crossing elections? *Autonomous Agents and Multiagent Systems*, 31(3):606–627, 2017.

[Mirrlees, 1971] J. Mirrlees. An exploration in the theory of optimal income taxation. *Review of Economic Studies*, 38:175–208, 1971.

[Obraztsova and Elkind, 2011] S. Obraztsova and E. Elkind. On the complexity of voting manipulation under randomized tie-breaking. In *Proceedings of IJCAI-11*, pages 319–324, July 2011.

[Obraztsova *et al.*, 2011] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *Proceedings of AAMAS-11*, pages 71–78, May 2011.

[Peters and Elkind, 2016] D. Peters and E. Elkind. Preferences single-peaked on nice trees. In *Proceedings of AAAI-16*, pages 594–600, 2016.

[Peters and Lackner, 2017] D. Peters and M. Lackner. Preferences single-peaked on a circle. In *Proceedings of AAAI-2017*, pages 649–655, 2017.

[Procaccia *et al.*, 2008] A. Procaccia, J. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[Roberts, 1977] K. Roberts. Voting over income tax schedules. *Journal of Public Economics*, 8(3):329–340, 1977.

[Rothe *et al.*, 2003] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner problem for Young elections. *Theory of Comp. Systems*, 36(4):375–386, 2003.

[Skowron *et al.*, 2015a] P. Skowron, P. Faliszewski, and A. Slinko. Achieving fully proportional representation: Approximability result. *Artificial Intelligence*, 222:67–103, 2015.

[Skowron *et al.*, 2015b] P. Skowron, L. Yu, P. Faliszewski, and E. Elkind. The complexity of fully proportional representation for single-crossing electorates. *Theoretical Computer Science*, 569:43–57, 2015.

[Young, 1977] H. Young. Extending Condorcet's rule. *Journal of Economic Theory*, 16(2):335–353, 1977.

[Yu *et al.*, 2013] Lan Yu, Hau Chan, and Edith Elkind. Multiwinner elections under preferences that are single-peaked on a tree. In *Proceedings of IJCAI-13*, volume 13, pages 425–431, 2013.