Voting in Two-Crossing Elections

Andrei Constantinescu and **Roger Wattenhofer**
ETH Zürich
{aconstantine, wattenhofer}@ethz.ch

Abstract

We introduce two-crossing elections as a generalization of single-crossing elections, showing a number of new results. First, we show that two-crossing elections can be recognized in polynomial time, by reduction to the well-studied consecutive ones problem. Single-crossing elections exhibit a transitive majority relation, from which many important results follow. On the other hand, we show that the classical Debord-McGarvey theorem can still be proven two-crossing, implying that any weighted majority tournament is inducible by a two-crossing election. This shows that many voting rules are NP-hard under two-crossing elections, including Kemeny and Slater. This is in contrast to the single-crossing case and outlines an important complexity boundary between single- and two-crossing. Subsequently, we show that for two-crossing elections the Young scores of all candidates can be computed in polynomial time, by formulating a totally unimodular linear program. Finally, we consider the Chamberlin–Courant rule with arbitrary disutilities and show that a winning committee can be computed in polynomial time, using an approach based on dynamic programming.

1 Introduction

Many impossibility results in social choice theory disappear if we assume restrictions on the voting preferences. The single-crossing domain is among the most studied restrictions in the literature. Not only does it make many social choice problems tractable, but it is also justifiable practically when placing both voters and candidates on a one-dimensional "left-right" spectrum. However, this one-dimensional model is often too restrictive, with real voting preferences hardly ever adhering to the single-crossing model. In this paper we wonder to what degree we can assume a more general model whilst preserving some of the theoretical and algorithmic properties of single-crossingness.

In multi-party democracies, one can often witness situations in which both the far left and the far right oppose against a change brought up by centrist parties. This was prominently happening during the Weimar Republic, but is common also today and known as "unholy alliance" or "The enemy of my enemy is my friend." This generalized political model is known as the horseshoe theory; if the far left and the far right are actually closer to each other than to the center, the political line bends into a horseshoe, or even a circle.

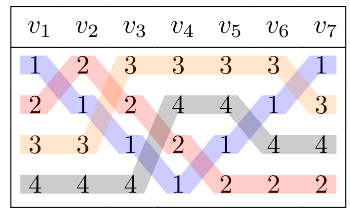

Figure 1: Two-crossing profile with 7 voters and 4 candidates. Each column, corresponding to a voter, lists alternatives in decreasing order of preference. Voters are given in a two-crossing order; i.e. any two of the four colored candidate trajectories cross at most twice.

An election is single-crossing if voters can be ordered into a list such that, as we sweep from left to right, the relative order between every pair of distinct candidates changes at most once. In this paper, we generalize the single-crossing property to a two-crossing property, such that the relative order of every pair of candidates is allowed to change at most twice (Figure 1). As we shall see, this can directly model a horseshoe political system. In contrast to single-crossing preferences, there is evidence that two-crossing preferences can accurately capture real voting behavior, particularly in economic preferences [Chen *et al.*, 2021]. It is worth noting that single-crossing preferences were also first introduced in similar economic settings [Mirrlees, 1971; Roberts, 1977]. Beyond these practical considerations, we believe that it is worth studying more general domain restrictions, and two-crossing seems like a natural candidate.

Determining the winners of an election is perhaps the oldest and most fundamental problem in the field. In most single-winner voting systems, finding the winners is straightforward, with a few notable exceptions, for which the problem is at least NP-hard: Young [Rothe *et al.*, 2003; Brandt *et al.*, 2015; Fitzsimmons and Hemaspaandra, 2020], Dodgson [Bartholdi *et al.*, 1989; Hemaspaandra *et al.*, 1997] and Kemeny [Bartholdi *et al.*, 1989; Hemaspaandra *et al.*, 2005]. For single-crossing preferences, all the aforementioned admit polynomial time algorithms, hinging essentially on the existence of (weak) Condorcet winners, which are easy to compute and, at least for an odd number of voters, actually give the winner straight away. For two-crossing elections, the situation is more interesting, as we shall see.

When it comes to multi-winner rules, there are many prominent examples which are NP-hard, including one of the most studied, the Chamberlin–Courant rule for proportional representation, which is hard even when voters' dissatisfaction values follow very simple patterns [Procaccia *et al.*, 2008; Lu and Boutilier, 2011]. On the other hand, computing a winning committee is easy for single-crossing preferences [Skowron *et al.*, 2015; Constantinescu and Elkind, 2021], but NP-hard for three-crossing preferences [Misra *et al.*, 2017]. Studying the two-crossing case is required to close this gap.

Our Contribution. We study two-crossing elections from an axiomatic and algorithmic point of view. Axiomatically, we show that they are equally expressive to unrestricted elections in terms of the (weighted) majority tournament: all weighted majority tournaments with same-parity weights are inducible by a two-crossing profile. Consequently, Slater, Banks, Minimal Extending Set, Tournament Equilibrium Set, Kemeny and Ranked Pairs are all NP-hard under two-crossing elections. Algorithmically, we show how recognition can be achieved in polynomial time and study the winner determination problem for Young's and the Chamberlin–Courant rule, in both cases providing polynomial time algorithms.

Full Version. The full version [Constantinescu and Wattenhofer, 2022] provides the omitted proofs and a discussion of recognizing k-crossingness for general k.

2 Preliminaries

Given integer n, we write $[n]$ for the set $\{1, \dots, n\}$; given two integers $\ell \le r$, we write $[\ell : r]$ to denote the set $\{\ell, \dots, r\}$. For a fixed n and $1 \le \ell < r \le n$ we write $[r : \ell]$ for the set $[1 : \ell] \cup [r : n]$. A subset $\mathcal{I} \subseteq [n]$ is an *interval* if $\mathcal{I} = [\ell : r]$ for some $1 \le \ell \le r \le n$ and a *circular interval* if we also allow for $\ell > r$ in the preceding condition. For a statement S, we write $[S]$ for the Iverson bracket: $[S] = 1$ if S holds, and 0 otherwise.

We consider a setting where voters $V = [n]$ express their preferences over a set of candidates (or alternatives) $C = [m]$. Voters rank candidates from best to worst, so that the preferences of a voter v are given by a linear order $\succ_v$: given two distinct candidates $c, c' \in C$ we write $c \succ_v c'$ when v prefers c to c'. The list of all voters' preferences is denoted by $\mathcal{P} = (\succ_v)_{v \in V}$, and is referred to as a *preference profile*. We use $\mathcal{L}(C)$ to denote the set of all linear orders over C, such that $\mathcal{P} \in \mathcal{L}(C)^n$.

Pairwise Majority and Young's Rule. Given a profile $\mathcal{P}$ over candidate set C and two candidates $c, c' \in C$ we write $n_{c,c'} = |\{i \in V : c \succ_i c'\}|$ for the number of voters who prefer c to c'; the so-called *majority margin* is then $m_{c,c'} = n_{c,c'} - n_{c',c}$. Consider orienting the edges of the complete graph on C such that $c \to c'$ iff $m_{c,c'} > 0$; leaving out edges with $m_{c,c'} = 0$. This construction is known as the *majority tournament* of election $\mathcal{P}$. If each edge $c \to c'$ is also given weight $m_{c,c'}$, then the resulting construction is known as the *weighted majority tournament* of $\mathcal{P}$. If there is a candidate $c \in C$ such that $m_{c,c'} > 0$ for all $c' \neq c$, then c is called the *strong Condorcet winner*. If the previous condition is weakened to $m_{c,c'} \ge 0$, then c is only a *weak Condorcet winner*, which there can be multiple of. For profile $\mathcal{P}$, the *strong (weak) Young score* of a candidate $c \in C$ is the minimum number of voters that need to be removed from $\mathcal{P}$ such that c becomes a strong (weak) Condorcet winner; in some cases this score will be infinite. In the strong (weak) *Young's rule* candidates with the smallest strong (weak) Young score are declared the winners. Traditionally, "Condorcet winner" means the strong variant, while "Young's rule" means the weak variant [Caragiannis *et al.*, 2016].

Tournament Solutions. Under some voting rules, knowledge of the (weighted) majority tournament is enough to compute the winners; we call such rules *(weighted) tournament solutions*. Some tournament solutions allow for polynomial time winner determination, the most natural example being Copeland. However, for a significant number of tournament solutions it is NP-hard to determine the winners — unweighted examples: Slater, Banks, Minimal Extending Set, Tournament Equilibrium Set; weighted examples: Kemeny, Ranked Pairs. See [Brandt *et al.*, 2016; Fischer *et al.*, 2016] for a survey on the topic.

In what follows, we assume that we are also given a *misrepresentation function* $\rho : V \times C \to \mathbb{Q}$; ρ is *consistent* with $\mathcal{P}$ if $c \succ_v c'$ implies $\rho(v, c) \le \rho(v, c')$ for all $v \in V$ and $c, c' \in C$. Intuitively, the value $\rho(v, c)$ indicates to what extent candidate c misrepresents voter v. A common example of a misrepresentation function is the *Borda function* ρ_B given by $\rho_B(v, c) = |\{c' \in C : c' \succ_v c\}|$; this function assigns value 0 to a voter's top choice, value 1 to his second choice, and value $m - 1$ to his last choice.

The Chamberlin–Courant Rule. A *multiwinner voting rule* maps a profile $\mathcal{P}$ over a candidate set C, and a positive integer $k \le |C|$, to a non-empty collection of subsets of C of size at most k;[1] the elements of this collection are called the *winning committees*. An *assignment function* is a mapping $w : V \to C$; for each $V' \subseteq V$ we write $w(V') = \{w(v) : v \in V'\}$ for the image of V' under w. If $|w(V)| \le k$, then w is called a *k-assignment*. Given a misrepresentation function ρ and a profile $\mathcal{P} = (\succ_v)_{v \in V}$, the *total dissatisfaction of voters in V under a k-assignment* w is given by $\Phi_\rho(\mathcal{P}, w) = \sum_{v \in V} \rho(v, w(v))$. Intuitively, $w(v)$ is the representative of voter v in the committee $w(V)$, and $\Phi_\rho(\mathcal{P}, w)$ measures to what extent the voters are dissatisfied with their representatives. An *optimal k-assignment* for ρ and $\mathcal{P}$ is a k-assignment that minimizes $\Phi_\rho(\mathcal{P}, w)$ among all k-assignments for $\mathcal{P}$. The *Chamberlin–Courant multiwinner voting rule* [Chamberlin and Courant, 1983; Faliszewski *et al.*, 2017] maps each triple $(\mathcal{P}, \rho, k)$ consisting of a preference profile $\mathcal{P} = (\succ_v)_{v \in V}$ over a candidate set C, a misrepresentation function $\rho : V \times C \to \mathbb{Q}$ that is consistent with $\mathcal{P}$, and a positive integer $k \le |C|$, to all sets W such that $W = w(V)$ for some k-assignment w that is optimal for ρ and $\mathcal{P}$, constituting the winning committees. It is

[1] Usually exactly k, but in our case the difference is immaterial.

NP-hard to determine whether a k-assignment of dissatisfaction $\Phi_\rho(\mathcal{P}, w) \leq B$ exists for some input parameter B, even in the special cases where $\rho(v, c) \in \{0, 1\}$ [Procaccia *et al.*, 2008], or if ρ is the Borda function [Lu and Boutilier, 2011].

k-Crossing Preferences. A profile $\mathcal{P} = (\succ_v)_{v \in V}$ over C is *k-crossing* if there is a permutation $(\sigma_i)_{i \in V}$ of V such that for every pair of distinct candidates $(c, c') \in C^2$ the number of indices $i \in [n-1]$ such that $[c \succ_{\sigma_i} c'] \neq [c \succ_{\sigma_{i+1}} c']$ is at most k. That is, if we order the voters in V according to σ and traverse the list of voters from left to right, each pair of candidates 'crosses' at most k times. In this case we write that $\mathcal{P}$ is k-C, with respect to σ. A profile $\mathcal{P}$ is *single-interval on a circle* if there is a permutation $(\sigma_i)_{i \in V}$ such that for all pairs of distinct candidates $(c, c') \in C^2$ the set $\{i \in V : c \succ_{\sigma_i} c'\}$ is a circular interval over $[n]$. If so, we write that $\mathcal{P}$ is $\mathrm{SI}_\bigcirc$ on the circle induced by σ. For $\Phi \in \{k\text{-C}, \mathrm{SI}_\bigcirc\}$ the *recognition* problem Φ^{REC} asks: given a preference profile $\mathcal{P}$, is there a permutation σ such that $\mathcal{P}$ is $\mathcal{L}$ with respect to σ? If yes, one is also interested in finding such a witnessing σ.

Lemma 1. *For any permutation $(\sigma_i)_{i \in V}$, profile $\mathcal{P}$ is two-crossing with respect to σ iff $\mathcal{P}$ is single-interval on the circle induced by σ. Consequently, $\mathcal{P}$ is two-crossing iff $\mathcal{P}$ is* $\mathrm{SI}_\bigcirc$.

Proof. Consider two candidates $c \neq c'$. If $\{i \in V : c \succ_{\sigma_i} c'\}$ is a circular interval, then there are at most two crosses, one at each interval end. Conversely, assume there are two crosses (the cases with one/zero crosses are similar) at positions $1 \leq i_1 < i_2 < n$ and that, without loss of generality, σ_1 prefers c to c'. Then, voters in $[i_1 + 1 : i_2]$ prefer c' to c and voters in $[i_2 + 1 : i_1]$ prefer c to c', both being circular intervals. $\square$

Lemma 2. *Consider a horseshoe political system: voters and candidates are assigned points on the unit circle, voters rank candidates in increasing order of circle arc-length distance from their assigned point, assume no ties. Then, voters' preferences are two-crossing.*

The Consecutive Ones Problem. An $n \times m$ binary matrix M has the *consecutive ones property* if its rows can be permuted such that in each column ones form a single continuous run. In this case, we say that M is C1P. Generalizing, M is k-C1P if its rows can be permuted such that in each column ones form at most k continuous runs. Similarly, M has the *circular consecutive ones property* if its rows can be permuted such that in each column ones form a single continuous run if we allow loop-around; we use $\mathrm{C1P}_\bigcirc$ to denote this property. For $\Phi \in \{\mathrm{C1P}, k\text{-C1P}, \mathrm{C1P}_\bigcirc\}$ we use Φ^{REC} to denote the corresponding recognition problems. It is instructive to note the following facts tying C1P and $\mathrm{C1P}_\bigcirc$.

Lemma 3 ([Tucker, 1971, Theorem 1])**.** *From a binary matrix M construct M^c by flipping ones and zeros on those columns with a one in the first row of M. Then, any permutation of rows σ fixing the first row witnesses that M is* $\mathrm{C1P}_\bigcirc$ *iff σ witnesses that M^c is* C1P.

Corollary 4. $\mathrm{C1P}_\bigcirc^{\mathrm{REC}}$ *and* $\mathrm{C1P}^{\mathrm{REC}}$ *are equivalent (interreducible with respect to complexity-preserving reductions). This also extends to finding witnessing permutations.*

Recognizing C1P/$\mathrm{C1P}_\bigcirc$ matrices and finding witnessing permutations can both be achieved in time linear in the size of the matrix [Booth and Lueker, 1976]. A simpler $O(nm^2)$ algorithm was given by [Fulkerson and Gross, 1965]. For $k \geq 2$, k-$\mathrm{C1P}^{\mathrm{REC}}$ is NP-complete [Goldberg *et al.*, 1995].

3 Recognizing k-Crossing Profiles

The problem of recognizing single-crossing elections admits a number of polynomial time algorithms [Elkind *et al.*, 2012; Bredereck *et al.*, 2013], but, to the best of our knowledge, recognizing k-crossing profiles for $k > 1$ has not been studied. We prove that recognizing two-crossing profiles reduces to recognizing C1P matrices, which is tractable in poly-time.

Given a profile $\mathcal{P}$ over candidate set C, let $M_\mathcal{P}$ be a binary matrix with n rows and $m(m-1)$ columns: one row for each voter $v \in V$ and one column for each pair of distinct candidates $(c, c') \in C^2$; such that $M_\mathcal{P}[v, (c, c')] = [c \succ_v c']$.

Lemma 5. *For any permutation $(\sigma_i)_{i \in V}$, $\mathcal{P}$ is single-interval on the circle induced by σ iff $M_\mathcal{P}$ is* $\mathrm{C1P}_\bigcirc$ *with respect to σ. Consequently, $\mathcal{P}$ is single-interval on a circle iff $M_\mathcal{P}$ is* $\mathrm{C1P}_\bigcirc$.

Proof. Consider two candidates $c \neq c'$. The requirement on (c, c') for $\mathrm{SI}_\bigcirc$ on the circle induced by σ to hold is that the set $\{i \in V : c \succ_{\sigma_i} c'\}$ is a circular interval over $[n]$. This is equivalent to saying that its indicator function $[c \succ_{\sigma_i} c']$ has all ones grouped into a single circular run, which is, by definition of $M_\mathcal{P}$, the same as saying that column (c, c') of $M_\mathcal{P}$ reordered by σ has all ones in a single circular interval. Quantifying over all $c \neq c'$ gives the conclusion. $\square$

Theorem 6. *Given a preference profile $\mathcal{P}$, deciding whether it is two-crossing and, if affirmative, finding a witnessing permutation can be done in time $O(nm^2)$.*

We note that the construction for $M_\mathcal{P}$ was first suggested by [Bredereck *et al.*, 2013], but their result is somewhat different: they show that $\mathcal{P}$ is single-crossing iff $M_\mathcal{P}$ is C1P, while we show that $\mathcal{P}$ is two-crossing iff $M_\mathcal{P}$ is $\mathrm{C1P}_\bigcirc$ (i.e. $M_\mathcal{P}^c$ is C1P). The relationship between consecutive ones and k-crossingness seems to go deeper than this. Say $\mathcal{P}$ is k-interval on a circle if voters can be rearranged into a circle such that for all pairs of distinct candidates $(c, c') \in C^2$ the set $\{i \in V : c \succ_i c'\}$ is a union of at most k circular intervals. Similarly, say a matrix is k-$\mathrm{C1P}_\bigcirc$ if its rows can be rearranged such that each column has at most k circular runs of ones. With these conventions, one can follow the proof of Lemma 1 to also show that $\mathcal{P}$ is $2k$-crossing iff $\mathcal{P}$ is k-interval on a circle. Moreover, one can follow the proof of Lemma 5 to show that $\mathcal{P}$ is k-interval on a circle iff $M_\mathcal{P}$ is k-$\mathrm{C1P}_\bigcirc$.

4 Weighted Majority Tournaments

Single-crossing is attractive from an axiomatic standpoint: the majority relation is transitive and a Condorcet winner exists, for odd n. The Condorcet Paradox profile consists of 3 voters, so it can be easily seen as two-crossing. Therefore, Condorcet winners might not exist under two-crossing. This begs the question, can we guarantee anything about the weighted majority tournament of a two-crossing election?

The following result, similar to [Peters and Lackner, 2020] for elections single peaked on a circle, answers negatively.

Theorem 7 (Debord-McGarvey for Two-Crossing Elections)**.** *Any weighted majority tournament with weights of the same parity*[2] *is inducible by a two-crossing election. If W denotes the maximum weight of an edge, then $O(m^2W)$ voters suffice.*

Proof. For some number of candidates m, consider the following "Double-BubbleSort" construction of a two-crossing profile. There will be $m(m-1)+1$ voters; voter 1 ranks $1 \succ 2 \succ \ldots \succ m$, which we more succinctly represent as the permutation $123\ldots m$. Voter 2 ranks $213\ldots m$, voter 3 ranks $231\ldots m$, and so on, voter m ranks $23\ldots m1$. In essence, one swap at a time, candidate 1 went from best to worst. For the following $m-1$ voters, candidate 2 will go from best to second worst, one position at a time: $324\ldots m1$, $342\ldots m1$, $\ldots$, $34\ldots m21$. In the following rounds, we similarly bring from front to back candidates $3, 4, \ldots m-1$, each taking multiple swaps to reach their final position. Note that the swaps we do are precisely those done by a BubbleSort algorithm sorting in descending order. This construction is illustrated for $m=4$ by the first 7 voters in Figure 2. To complete the profile, we need $m(m-1)/2$ additional voters; consecutive voters will, once again, differ by a single adjacent swap. In particular, we go from $m\ldots 21$ back to $12\ldots m$ by following the swaps of a BubbleSort algorithm sorting in ascending order, but iterating over the permutation in reverse order. Figure 2, voters v_7 to v_{13}, demonstrates this process. Note that the majority margins satisfy $m_{c,c'} = 1$ for $c < c'$.

Why is this profile interesting? For any two candidates $c \neq c'$ there are voters v_i, v_j with preference permutations $\tau_i = Acc'B$ and $\tau_j = \overline{B}cc'\overline{A}$; where $\overline{X}$ denotes the reverse of permutation X. This means that, if we add one additional copy of voters v_i, v_j to our profile, then all majority margins stay unchanged, except for $m_{c,c'}$ and $m_{c',c}$, which increase/decrease by 2. Since c, c' were arbitrary, this means that we can increase/decrease any majority margin by 2 without affecting the others,[3] by adding two additional voters. If weights are odd, this can be done repeatedly until our profile has margins agreeing to the weights in the tournament. Fixing the margin for a weight $w \leq W$ requires $O(w)$ additions of voters, so overall we need at most $O(m^2W)$ voters. The case of even weights is similar: add an additional $m\ldots 21$ voter to our original profile, and then proceed analogously. $\square$

The previous result shows that, essentially, the computational complexity of all known (weighted) tournament solutions is unchanged from the general case by restricting to two-crossing elections. Thus, we have the following.

Corollary 8. *Determining the winners for Slater, Banks, Minimal Extending Set, Tournament Equilibrium Set, Kemeny and Ranked Pairs is NP-hard under two-crossing elections.*

[2]Nonexistent edges are considered to have weight 0.

[3]Subject to the implicit constraint that $m_{c,c'} + m_{c',c} = 0$.

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}
1	2	2	2	3	3	4	4	4	1	1	1	1
2	1	3	3	2	4	3	3	1	4	4	2	2
3	3	1	4	4	2	2	1	3	3	2	4	3
4	4	4	1	1	1	1	2	2	2	3	3	4

Figure 2: Construction in the proof of Theorem 7 for $m = 4$ candidates. Voters are arranged in a two-crossing order.

5 Young's Rule

In this section we show how Young scores can be computed in polynomial time by setting up integer programs whose LP relaxations have totally unimodular matrices. This means that vertices of the feasible regions have integer coordinates, so it is enough to solve the LPs.

Theorem 9. *Given a two-crossing profile $\mathcal{P}$ with voter set V and candidate set C, and a candidate $c \in C$, the weak and strong Young scores of c can be computed in polynomial time.*

Proof. We present the algorithm for the weak Young score, the strong case being analogous. We are hence interested in removing a minimal number of voters such that the majority margins satisfy $m_{c,c'} \geq 0$ for all $c' \in C \setminus \{c\}$. To do so, set up an integer program P with variables $(x_v)_{v \in V} \in \{0,1\}$, where x_v is 1 iff voter v is kept; i.e. it is not removed. Then, $m_{c.c'} \geq 0$ can be rewritten as $\sum_{v \in V} x_v \llbracket c \succ_v c' \rrbracket \geq 0$; where by $\llbracket S \rrbracket = 2[S] - 1$ we mean 1 when S holds and -1 otherwise. The goal is to maximize $\sum_{v \in V} x_v$. If we ignore the $x_v \in \{0,1\}$ constraints and henceforth assume that voters are ordered in a two-crossing fashion, then the constraints matrix of P satisfies a property similar to the row-wise version (implicit from now on) of C1P$_\bigcirc$: it consists of values ± 1 and on each row ones form a circular interval. Unfortunately, not all such matrices are totally unimodular; e.g the 2×2 matrix with 1 on the main diagonal and -1 otherwise.

To mitigate this difficulty, we need a slightly different approach: we will successively check for each value $s = n, n-1, \ldots, 0$ whether a solution keeping exactly s of the n voters exists. For a fixed s, we add the additional constraint that $\sum_{v \in V} x_v = s$. With the sum of the x's fixed we can rewrite our main inequalities as follows:

$$\sum_{v \in V} x_v \llbracket c \succ_v c' \rrbracket \geq 0 \text{ iff } \sum_{v \in V} x_v \left(2\left[c \succ_v c'\right] - 1\right) \geq 0 \text{ iff}$$
$$2\sum_{v \in V} x_v \left[c \succ_v c'\right] \geq \sum_{v \in V} x_v \text{ iff } \sum_{v \in V} x_v \left[c \succ_v c'\right] \geq \left\lceil \frac{s}{2} \right\rceil$$

Therefore, for a fixed s, our integer program P_s consists of checking the feasibility of the following constraints:

$$\sum_{v \in V} x_v = s \tag{1}$$

$$\sum_{v \in V} x_v \left[c \succ_v c'\right] \geq \left\lceil \frac{s}{2} \right\rceil, \qquad c' \in C \setminus \{c\} \tag{2}$$

$$x_v \in \{0,1\}, \qquad v \in V \tag{3}$$

At this point, if we relax constraints (3) to $0 \leq x_v \leq 1$, then the matrix of the resulting linear program L_s satisfies C1P$_\bigcirc$. Unfortunately, this still does not guarantee total unimodularity; e.g. the 3×3 matrix with 0 on the main diagonal and 1 otherwise has determinant $2 \notin \{-1, 0, 1\}$.

In one last step, we slightly adjust our constraints to get a matrix which is C1P, and hence provably totally unimodular [Fulkerson and Gross, 1965]. Note that constraints (1) and (3) do not make use of circularity, so only constraints (2) might need adjustments. Consider some constraint $\sum_{v \in V} x_v \,[c \succ_v c'] \geq \left\lceil \frac{s}{2} \right\rceil$ needing adjustment; this is because $[c \succ_v c']$ is of the form $11 \dots 1100 \dots 0011 \dots 11$ as v ranges over the voters. We can then write

$$\sum_{v \in V} x_v \,[c \succ_v c'] \geq \left\lceil \frac{s}{2} \right\rceil \text{ iff } \sum_{v \in V} x_v \,(1 - [c' \succ_v c]) \geq \left\lceil \frac{s}{2} \right\rceil \text{ iff}$$

$$s - \sum_{v \in V} x_v \,[c' \succ_v c] \geq \left\lceil \frac{s}{2} \right\rceil \text{ iff } \sum_{v \in V} x_v \,[c' \succ_v c] \leq \left\lfloor \frac{s}{2} \right\rfloor$$

For this equivalent condition, the coefficients $[c' \succ_v c]$ are of the form $00 \dots 0011 \dots 1100 \dots 00$ as v ranges over V. Therefore, if we replace each constraint which uses circularity as such, the matrix of the resulting LP will be C1P, and so totally unimodular. This means that the feasible region has integer vertices [Hoffman and Kruskal, 2016], so the IP and the LP are equifeasible. Note that the replacements we made in (2) proved total unimodularity, but need not be executed in practice; i.e. it is enough to check the feasibility of L_s. If we now use any polynomial time algorithm for checking feasibility, like [Karmarkar, 1984], the poly-time bound follows. $\square$

Theorem 10. *Given a two-crossing profile $\mathcal{P}$ and $c \in C$, the strong/weak Young score of c can be computed in time $O((n+m^2)n^{3/2} \log n)$ by further developing our techniques.*

6 The Chamberlin–Courant Rule

In this section we show that computing the least possible dissatisfaction and also a winning committee for the Chamberlin–Courant rule (CC in this section) can both be achieved in polynomial time under two-crossing elections. Since the recognition problem can be solved in polynomial time by Theorem 6, assume voters $V = [n]$ are ordered such that our profile is two-crossing with respect to the identity permutation. For the following key lemma, we call a k-assignment function w *illegal* if there are two candidates $a \neq b$ and four voters $i_1 < i_2 < i_3 < i_4$ such that $w(i_1) = w(i_3) = a$ and $w(i_2) = w(i_4) = b$. We call w *legal* if it is not illegal.

Lemma 11. *For any instance $(\mathcal{P}, \rho, k)$ of CC there exists a legal k-assignment w_{opt} that is optimal for ρ and $\mathcal{P}$.*

Proof. Start with any optimal k-assignment w. If w is legal, then we are done, otherwise let $a \neq b$ and $i_1 < i_2 < i_3 < i_4$ witnesses the illegality of w. Since $\mathcal{P}$ is two-crossing we know it can not be the case that $[a \succ_{i_1} b] = [a \succ_{i_3} b] \neq [b \succ_{i_2} a] = [b \succ_{i_4} a]$, as that would result in 3 crosses for the pair (a, b). Therefore, for at least one of i_1, i_2, i_3, i_4 assignment w assigns a candidate from $\{a, b\}$ which is less preferred than the other, so just exchanging a for b, or vice-versa, for that voter results in an assignment w' such that $\Phi_\rho(\mathcal{P}, w') \leq \Phi_\rho(\mathcal{P}, w)$. Since w was optimal, it follows that w' is also optimal. We can now replace w by w' and repeat the reasoning iteratively, until we eventually reach a legal optimal k-assignment w_{opt}. It remains to show that this process terminates, but this is easy to see since at each step we replace a voter's assigned candidate with one that is strictly more preferred by them. $\square$

To find a winning committee for CC it is enough to find an optimal k-assignment for ρ and $\mathcal{P}$. Given Lemma 11, we can limit our search to legal k-assignments. It turns out that legal k-assignments admit a second characterisation, as follows. The proof is straightforward, so we leave it to the reader.

Proposition 12. *A k-assignment $w = w(1), w(2), \dots, w(n)$ is legal iff for any candidates $a \neq b$ there are no a's between the first and the last b or no b's between the first and last a.*

Given this, any legal assignment w induces a strict partial order relation $\to^*$ over C, where $a \to^* b$ for $a \neq b$ iff there are b's between the first and the last a in w. Note that this implicitly requires no a's between the first and the last b, by Proposition 12. Observe that two candidates $a \neq b$ are incomparable with respect to $\to^*$ iff all a's precede all b's in w, or vice-versa. Relation $\to^*$ satisfies an additional property: if $a \neq b$ are incomparable, then there can be no $c \notin \{a, b\}$ such that $a \to^* c$ and $b \to^* c$, since a structure of the form $a \dots c \dots a \dots b \dots c \dots b$ occurring in w would be illegal for pairs (a, c) and (b, c). These being said, one can now observe that the covering relation $\to$ of $\to^*$; i.e. the inclusion minimal relation with the same transitive closure; is a forest of directed trees.[4] Since our partial order $\to^*$ is finite, there exists at least one maximal element $c_{rt} \in C$, which we call a "root" element; i.e. such that there is no $c \in C$ with $c \to^* c_{rt}$. Let us now analyze w with respect to c_{rt}. Since c_{rt} is maximal, for all $c \in C \setminus \{c_{rt}\}$, no c_{rt}'s occur between the first and the last c in w. In other words, if we consider the set $w^{-1}(c_{rt})$ of positions where c_{rt} occurs in w, then all c's occur between two consecutive c_{rt}'s, or before/after the first/last c_{rt}. Put differently, candidate c_{rt} "splits" the range $1 \dots n$ into buckets delimited by consecutive entries in $w^{-1}(c_{rt}) \cup \{0, n+1\}$, such that any candidate $c \in C \setminus \{c_{rt}\}$ appears in at most one bucket. This reasoning can then be carried out recursively in each bucket as well.

The idea of our approach can now be stated intuitively: we will proceed by dynamic programming, solving the main problem by trying out all possible values of $w^{-1}(c_{rt})$, and then invoking recursively on each resulting bucket. There are a number of burning issues at this point: (i) there are exponentially many values of $w^{-1}(c_{rt})$ to try, (ii) there are exponentially many ways to distribute the number k of representatives into the buckets (iii) how do we ensure that no candidate is used in two distinct buckets, maybe further down the recursion, fact which might render the assignment illegal? Before addressing (i)-(ii) specifically, let us first give a preliminary version of the dynamic program, which will run in exponential time, and argue for its correctness in addressing (iii). Introduce $dp[\ell, r, t]$ for $1 \leq \ell \leq r \leq n$, $1 \leq t \leq k$

[4] We do not use this fact explicitly, but it helps gain intuition for the DP which follows.

to mean the least total dissatisfaction attainable for voters in $[\ell : r]$ using a legal t-assignment restricted to those voters. For ease of presentation, we use the degenerate base-cases $dp[\ell, \ell - 1, -] = 0$ for $\ell \in [n + 1]$ and $dp[\ell, r, 0] = \infty$ for $1 \le \ell \le r \le n$. Then, the recurrence relation is:

$$dp[\ell, r, t] = \min \Bigg\{ \sum_{i=1}^{x} \rho(w_i, c_{rt}) + \sum_{i=0}^{x} dp[w_i + 1, w_{i+1} - 1, t_i] : c_{rt} \in C, 1 \le x \le r - \ell + 1, \ell \le w_1 < \ldots < w_x \le r, t_0, \ldots, t_x \ge 0 \text{ and } t_0 + \ldots + t_x = t - 1 \Bigg\} \tag{4}$$

where we wrote $w_1, \ldots, w_x$ for the elements of $w^{-1}(c_{rt})$ and used the convention that $w_0 = \ell - 1$ and $w_{x+1} = r + 1$. In other words: we choose the root candidate c_{rt}, we choose $w^{-1}(c_{rt})$, and we choose a way to distribute the remaining $t-1$ candidates among the buckets; each bucket is then solved "recursively" with the respective allowed committee size. For completeness, subproblems can be computed in increasing order of t, breaking ties arbitrarily. The cost of an optimal k-assignment w_{opt} will be in $dp[1, n, k]$ at the end. To retrieve one such w_{opt}, additional bookkeeping is required: for each subproblem (ℓ, r, x) keep track of which values of c_{rt}, x, $w_1, \ldots, w_x$ and $t_0, \ldots, t_x$ were used to achieve the minimum, and then at the end trace those back to build w_{opt}.

As argued previously, not only does this DP require exponential time to compute, it is not even clearly correct, reason being that it can produce illegal k-assignments by reusing candidates, especially along different branches of the recursion. However, note that this comes at a price: whenever a candidate is reused it is counted as a new candidate out of the maximum of k allowed. Nevertheless, none of this is harmful: consider the reconstructed optimal assignment w_{opt} returned by the DP. This is a k-assignment since we only ever over-count candidates in the assembly, but never under-count. By construction, our recurrence clearly correctly considers all legal k-assignments, and will thus return a cost at most that of the least dissatisfaction legal k-assignment. Since by Lemma 11 no illegal k-assignment can be better than the best legal one, it follows that the returned assignment w_{opt} is optimal, regardless of whether it is legal or not.

This proof approach can be summarized as follows, and is also implicit in the works of [Constantinescu and Elkind, 2021]: (1) show optimal solutions with a rigid structure exist; (2) set up a DP which looks only for solutions of this form; (3) note that the DP occasionally produces non-conforming solutions, but without affecting global correctness.

Having addressed concern (iii), we now need to speed up the DP — we need a faster way to compute (4). To achieve this, introduce an auxiliary second dynamic program, $dp_2[\ell, r, t, c_{rt}]$, with the same semantics as dp, but enforcing a certain value for the root c_{rt}. It is straightforward to see that

$$dp[\ell, r, t] = \min_{c_{rt} \in C} dp_2[\ell, r, t, c_{rt}] \tag{5}$$

so it remains to show how dp_2 can be computed in polynomial time, which we do with the following.

Lemma 13. *dp_2 satisfies the following recurrence relation:*

$$dp_2[\ell, r, t, c_{rt}] = \min \big\{ dp[\ell, w_1 - 1, t_0] + \rho(w_1, c_{rt}) + \min \{ dp[w_1 + 1, r, t - t_0 - 1], dp_2[w_1 + 1, r, t - t_0, c_{rt}] \} : w_1 \in [\ell, r], 0 \le t_0 \le t - 1 \big\} \tag{6}$$

Proof. The key stands in noting that it makes little sense to try out all values of $w^{-1}(c_{rt}) = \{w_1, \ldots\}$ at once, so let us only go through all values of $w_1 \in [\ell : r]$ and $t_0 \in [0 : t-1]$ and observe the optimal substructure: for the left part, i.e. range $[\ell : w_1 - 1]$, we use the optimal unrestricted solution with at most t_0 candidates, i.e. $dp[\ell, w_1 - 1, t_0]$, for position w_1 we take the fixed cost $\rho(w_1, c_{rt})$, while for the right part, i.e. range $[w_1 + 1, r]$, we have more choice. Namely, there are two possibilities: w_1 was the only element in $w^{-1}(c_{rt})$, in which case we need to take the best unrestricted solution with at most $t - t_0 - 1$ candidates, the -1 accounting for candidate c_{rt}, i.e. $dp[w_1 + 1, r, t - t_0 - 1]$, or $|w^{-1}(c_{rt})| > 1$, in which case there is a w_2 (and potentially more occurrences of c_{rt}) to be placed; doing so optimally is precisely the definition of $dp_2[w_1 + 1, r, t - t_0, c_{rt}]$.[5] Altogether, we get (6). ☐

We now outline our main result. Essentially, one can now compute subproblems in increasing order of $r - \ell$, the optimal dissatisfaction being in $dp[1, n, k]$ at the end.

Theorem 14. *Given an instance $(\mathcal{P}, \rho, k)$ of CC where $\mathcal{P}$ is two-crossing, the optimal dissatisfaction and some winning committee can be computed in polynomial time.*

We note that for "egalitarian" Chamberlin–Courant, where one is instead interested in minimizing the dissatisfaction of the most misrepresented voter, simply replacing '+' with `max` in the DPs preserves correctness.

7 Conclusions and Future Work

We investigated two-crossing elections, giving an efficient recognition algorithm. Axiomatically, we showed that two-crossing elections are no different from general elections for any voting rule operating on the (weighted) majority tournament, such as Kemeny and Slater. Subsequently, we considered Young and the Chamberlin–Courant rule. In both cases we gave polynomial algorithms which are applicable in practice. We leave open whether faster solutions exist.

So far, two-crossing elections have not received the attention they deserve in the social choice literature. Algorithmically, we leave open whether Dodgson winners can be computed in polynomial time under two-crossing elections. k-crossing preferences for $k > 2$ are strictly more expressive than their two-crossing counterparts, but this comes with an increase in computational complexity; e.g. computing a winning committee for the Chamberlin–Courant rule, while being in P for two-crossing elections, becomes NP-hard under three-crossing elections. It would be interesting to establish such tight bounds for other voting rules; e.g. Young's; and, perhaps most importantly, to establish the hardness of recognizing k-crossing elections for $k > 2$. Axiomatically, it would be interesting to determine whether two-crossing elections admit a forbidden structure characterization, similarly to single-crossing elections [Bredereck *et al.*, 2013].

[5]No -1 here because c_{rt} is still actively "in use" at this point.

Acknowledgements

We thank Edith Elkind for the many useful discussions about two-crossing elections. We thank Costin Andrei Oncescu for mentioning a technique useful for improving the polynomial running time for Young's rule.

References

[Bartholdi *et al.*, 1989] J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.

[Booth and Lueker, 1976] Kellogg S. Booth and George S. 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] Felix Brandt, Markus Brill, Edith Hemaspaandra, and Lane A. Hemaspaandra. Bypassing combinatorial protections: polynomial-time algorithms for single-peaked electorates. *J. Artif. Int. Res.*, 53(1):439–496, may 2015.

[Brandt *et al.*, 2016] Felix Brandt, Markus Brill, and Paul Harrenstein. Tournament solutions. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 3. Cambridge University Press, USA, 1st edition, 2016.

[Bredereck *et al.*, 2013] Robert Bredereck, Jiehua Chen, and Gerhard J. Woeginger. A characterization of the single-crossing domain. *Social Choice and Welfare*, 41(4):989–998, 2013.

[Caragiannis *et al.*, 2016] Ioannis Caragiannis, Edith Hemaspaandra, and Lane A. Hemaspaandra. Dodgson's rule and young's rule. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 5. Cambridge University Press, USA, 1st edition, 2016.

[Chamberlin and Courant, 1983] John Chamberlin and Paul Courant. Representative deliberations and representative decisions: proportional representation and the borda rule. *American Political Science Review*, 77(3):718–733, 1983.

[Chen *et al.*, 2021] Chia-Hui Chen, Junichiro Ishida, and Wing Suen. Signaling under Double-Crossing Preferences. ISER Discussion Paper 1103rr, Institute of Social and Economic Research, Osaka University, Oct 2021.

[Constantinescu and Elkind, 2021] Andrei Costin Constantinescu and Edith Elkind. Proportional representation under single-crossing preferences revisited. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):5286–5293, May 2021.

[Constantinescu and Wattenhofer, 2022] Andrei Costin Constantinescu and Roger Wattenhofer. Voting in two-crossing elections. *arXiv preprint*, arXiv:2205.00474, 2022.

[Elkind *et al.*, 2012] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. Clone structures in voters' preferences. *Proceedings of the ACM Conference on Electronic Commerce*, pages 496–513, 2012.

[Faliszewski *et al.*, 2017] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Multiwinner voting: a new challenge for social choice theory. In Ulle Endriss, editor, *Trends in Computational Social Choice*, chapter 2, pages 27–47. AI Access, 2017.

[Fischer *et al.*, 2016] Felix Fischer, Olivier Hudry, and Rolf Niedermeier. Weighted tournament solutions. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 4. Cambridge University Press, USA, 1st edition, 2016.

[Fitzsimmons and Hemaspaandra, 2020] Zack Fitzsimmons and Edith Hemaspaandra. Election score can be harder than winner. In *ECAI 2020*, 2020.

[Fulkerson and Gross, 1965] Delbert Ray Fulkerson and Oliver Alfred Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15:835–855, 1965.

[Goldberg *et al.*, 1995] Paul Goldberg, Martin Golumbic, Haim Kaplan, and Ron Shamir. Four strikes against physical mapping of dna. *Journal of computational biology: a journal of computational molecular cell biology*, 2:139–52, 03 1995.

[Hemaspaandra *et al.*, 1997] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Exact analysis of dodgson elections: lewis carroll's 1876 voting system is complete for parallel access to np. *J. ACM*, 44(6):806–825, Nov 1997.

[Hemaspaandra *et al.*, 2005] Edith Hemaspaandra, Holger Spakowski, and Jörg Vogel. The complexity of kemeny elections. *Theoretical Computer Science*, 349:382–391, 12 2005.

[Hoffman and Kruskal, 2016] A. J. Hoffman and J. B. Kruskal. Integral boundary points of convex polyhedra. In *Linear Inequalities and Related Systems. (AM-38), Volume 38*, chapter 13, pages 223–246. Princeton University Press, 2016.

[Karmarkar, 1984] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC'84, pages 302–311, New York, NY, USA, 1984. Association for Computing Machinery.

[Lu and Boutilier, 2011] Tyler Lu and Craig Boutilier. Budgeted social choice: from consensus to personalized decision making. In *Proceedings of IJCAI'11*, pages 280–286, 01 2011.

[Mirrlees, 1971] James Mirrlees. An exploration in the theory of optimum income taxation. *The Review of Economic Studies*, 38(2):175–208, 1971.

[Misra *et al.*, 2017] Neeldhara Misra, Chinmay Sonar, and P. R. Vaidyanathan. On the complexity of chamberlin–courant on almost structured profiles. In Jörg Rothe, editor, *Algorithmic Decision Theory*, pages 124–138, Cham, 2017. Springer International Publishing.

[Peters and Lackner, 2020] Dominik Peters and Martin Lackner. Preferences single-peaked on a circle. *Journal of Artificial Intelligence Research*, 68:463–502, 06 2020.

[Procaccia *et al.*, 2008] Ariel Procaccia, Jeffrey Rosenschein, and Aviv Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30:353–362, 02 2008.

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

[Rothe *et al.*, 2003] Jörg Rothe, Holger Spakowski, and Jörg Vogel. Exact complexity of the winner problem for young elections. *Theory of Computing Systems*, 36:375–386, 2003.

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

[Tucker, 1971] Alan C. Tucker. Matrix characterizations of circular-arc graphs. *Pacific Journal of Mathematics*, 39:535–545, 1971.