Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Multiwinner Elections Under Preferences that Are Single-Peaked on a Tree

Lan Yu[1], **Hau Chan**[2], and **Edith Elkind**[1]
[1]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
YULA0001@e.ntu.edu.sg, eelkind@ntu.edu.sg
[2]Department of Computer Science, Stony Brook University, USA
hauchan@cs.stonybrook.edu

Abstract

We study the complexity of electing a committee under several variants of the Chamberlin–Courant rule when the voters' preferences are single-peaked on a tree. We first show that this problem is easy for the egalitarian, or "minimax" version of this problem, for arbitrary trees and misrepresentation functions. For the standard (utilitarian) version of this problem we provide an algorithm for an arbitrary misrepresentation function whose running time is polynomial in the input size as long as the number of leaves of the underlying tree is bounded by a constant. On the other hand, we prove that our problem remains computationally hard on trees that have bounded degree, diameter, or pathwidth. Finally, we show how to modify Trick's [1989] algorithm to check whether an election is single-peaked on a tree whose number of leaves does not exceed a given parameter λ.

1 Introduction

Computational social choice is an active research area that deals with algorithmic aspects of collective decision-making. One of the most fundamental questions studied in this area is the complexity of determining the election winner(s) for various voting rules: indeed, for a rule to be practically applicable, it has to be the case that we can find the winner of an election in a reasonable amount of time.

Most common rules that are designed to output a single winner (possibly after an application of a tie-breaking rule) admit polynomial-time winner determination algorithms; the examples include such diverse rules as Plurality, Borda, Maximin, Copeland, and Bucklin (see, e.g., [Arrow *et al.*, 2002] for definitions). However, there are also some intuitively appealing single-winner rules for which winner determination is known to be computationally hard: this is the case, for instance, for Dodgson's rule [Bartholdi *et al.*, 1989; Hemaspaandra *et al.*, 1997], Young's rule [Rothe *et al.*, 2003], and Kemeny's rule [Hemaspaandra *et al.*, 2005]. More recently, there has been some interest in the computational complexity of voting rules whose purpose is to elect a representative assembly of candidates rather than select a single winner. While one can adapt common single-winner rules to this setting (e.g., appoint the candidates with the top k scores, where k is the target assembly size, or split the voters into k districts and determine the winner in each district using a single-winner rule), this approach may result in an assembly that does not reflect the true preferences of the electorate (see, e.g., [Betzler *et al.*, 2013]). Therefore, it is preferable to use a voting system that is specifically designed for multi-winner elections.

One such system was proposed by Chamberlin and Courant [1983]. It is based on the notion of a *misrepresentation function*—a mapping μ that, for every voter i and every candidate c, outputs a non-negative number $\mu(i, c)$, which indicates how badly c misrepresents i; typically, it is assumed that $\mu(i, c) = 0$ if c is i's most preferred candidate. Given an assembly A, a voter i is assumed to be represented by his most preferred candidate in A, i.e., an element of $\arg\min_{c\in A} \mu(i, c)$. There is no constraint on the number of voters that can be represented by a single candidate; the assumption is that the assembly will make its decisions by weighted voting, where the weight of each candidate is proportional to the fraction of the electorate that she represents. Chamberlin–Courant's scheme outputs an assembly of a given size that minimizes the *total misrepresentation*, interpreted as the sum of misrepresentations over all voters (see Section 2 for a formal definition). Monroe [1995] has subsequently proposed a variant of this scheme where the assembly is assumed to use non-weighted voting, and, consequently, each member of the assembly is required to represent approximately the same number of voters (up to a rounding error). Very recently, Betzler *et al.* [2013] suggested egalitarian, or "minimax", variants of both schemes, where the quality of the assembly is measured by the worst-case misrepresentation (taken over all voters) rather than total (or, equivalently, average) misrepresentation.

Unfortunately, the problem of identifying an optimal assembly under either Chamberlin–Courant's scheme or Monroe's scheme is known to be computationally hard, even for fairly simple misrepresentation functions. In particular, Procaccia *et al.* [2008] show that this is the case for both schemes under approval misrepresentation function (where μ takes values in $\{0, 1\}$) and Lu and Boutilier [2011] give an NP-hardness proof for Chamberlin–Courant's scheme under Borda misrepresentation function (where $\mu(i, c)$ is the number of candidates that i ranks above c). Betzler *et al.* [2013]

extend these hardness results to the egalitarian variants of both schemes.

Clearly, this is bad news if we want to use Chamberlin–Courant's scheme or Monroe's scheme in practice: elections may involve millions of voters and hundreds of candidates, and the election outcome needs to be announced soon after the votes have been cast. Thus, it is natural to try to circumvent the hardness results, either by designing efficient algorithms that compute an *approximately optimal* assembly or by identifying reasonable assumptions on the structure of the election that ensure computational tractability. The former approach was pursued by Lu and Boutilier [2011], and, more recently, by Skowron *et al.* [2013]. The latter approach was initiated by Betzler *et al.* [2013] who provide an extensive analysis of the fixed-parameter tractability of the winner determination problem under both utilitarian and egalitarian variants of Chamberlin–Courant's and Monroe's rules, as well as investigate the complexity of this problem for *single-peaked electorates*; Cornaz et al. [2012] extend the latter set of results to elections with bounded *single-peaked width*.

Recall that an election is said to be single-peaked [Black, 1958] if the voters' preferences over the candidates are determined by the candidates' positions with respect to a single one-dimensional issue such as the income tax or the military budget: each voter has his preferred position on this issue and ranks the candidates accordingly. Many voting-related problems that are known to be hard for the general electorates become easy when the voters' preferences are single-peaked. Betzler *et al.* [2013] show that this is also the case for winner determination under both variants (utilitarian and egalitarian) of Chamberlin–Courant's rule and for the egalitarian variant of Monroe's rule (for any misrepresentation function).

Our Contribution The goal of this paper is to explore whether the easiness results of Betzler *et al.* [2013] for single-peaked electorates can be extended to a more general class of elections, namely, those where the voters' preferences are *single-peaked on a tree*. Informally, an election belongs to this class if we can construct a tree whose vertices are candidates in the election, and each voter ranks all other candidates according to their perceived distance along this tree from his most preferred candidate. This generalization of the notion of single-peakedness was introduced by Demange [1982] and captures a much broader class of voters' preferences, while still implying the existence of a Condorcet winner.

We focus on Chamberlin–Courant's voting rule. We first show that for the egalitarian variant of this rule winner determination is easy for an arbitrary misrepresentation function as long as voters' preferences are single-peaked on a tree. Our proof proceeds by reducing our problem to an easy variant of the HITTING SET problem. For the utilitarian setting, we present an efficient winner determination algorithm for preferences that are single-peaked on a tree with a small number of leaves: the running time of our algorithm is polynomial in the size of the election, but exponential in the number of leaves (Section 4). We then argue (Section 5) that it is unlikely that this problem is easy for a richer class of trees. Specifically, we characterize elections that are single-peaked on a star (a tree with m vertices and $m-1$ leaves) and prove that for such elections it is NP-hard to determine the Chamberlin–Courant winners. We then modify our hardness reduction to show that our problem remains hard on trees of maximum degree 3. Thus, restricting the diameter, the pathwidth, or the maximum degree of the tree is insufficient to ensure the tractability of the winner determination problem for Chamberlin–Courant's rule.

The hardness results of Section 5 hold for a large class of misrepresentation functions, which, however, does not include the Borda misrepresentation function. Interestingly, we are able to show (Section 5.1) that for elections that are single-peaked on a star one can efficiently compute the Chamberlin–Courant winners with respect to the Borda misrepresentation function. It is not clear, however, if this easiness result can be extended to more general trees.

Now, the algorithm described in Section 4 assumes that the tree with respect to which the preferences are single-peaked is given as an input. However, in practice we cannot expect this to be the case: typically, we are only given the voters' preferences and have to construct such a tree (if it exists) ourselves. While the algorithm of Betzler *et al.* faces the same issue (i.e., it needs to know the societal axis), there exist efficient algorithms for determining the societal axis given the voters' preferences [Bartholdi and Trick, 1986; Escoffier *et al.*, 2008]. In contrast, for trees the situation is more complicated. Namely, Trick [1989] describes a polynomial-time algorithm that decides whether there exists a tree such that a given election is single-peaked with respect to it, and constructs *some* such tree if this is indeed the case. However, Trick's algorithm leaves us a lot of freedom when constructing the tree; as a result, if the election is single-peaked with respect to several different trees, the output of Trick's algorithm will be dependent on the implementation details. In particular, there is no guarantee that an arbitrary implementation will find a tree with a small number of leaves when it exists. Indeed, Trick [1989] acknowledges that his algorithm cannot be used to decide whether an election is single-peaked on a path. Fortunately, it turns out that we can implement Trick's algorithm so as to produce a tree with the smallest possible number of leaves; we describe the details of our implementation in Section 6. Thus, given an election that is single-peaked on some tree with at most λ leaves (where λ is a given constant), we can efficiently construct some such tree and pass it on to the winner determination algorithm described in Section 4; as a result, we can compute the Chamberlin–Courant winners of such elections in polynomial time.

2 Preliminaries

An *election* $E = (C, V)$ is given by a finite set of *candidates* $C = \{c_1, \dots, c_m\}$ and a set of *voters* $V = \{1, \dots, n\}$. Each voter $i \in V$ is associated with a *preference order* $\succ_i$, which is a total order over C; if $a \succ_i b$ for some $a, b \in C$, we say that voter i prefers candidate a to candidate b. We denote by $\mathrm{pos}(i, c)$ the position of candidate $c \in C$ in the preference order of voter $i \in V$: $\mathrm{pos}(i, c) = |\{c' \in C \mid c' \succ_i c\}| + 1$. Given a subset of candidates C' and a voter $i \in V$, we let $\mathrm{top}(i, C')$ and $\mathrm{bot}(i, C')$ denote voter i's most and least preferred candidate in C', respectively.

Multi-Winner Elections We denote by $\mathbb{Z}_+$ the set of all non-negative integers. A *misrepresentation function* for E is a mapping $\mu : V \times C \to \mathbb{Z}_+$ such that $\mathrm{pos}(i,c) < \mathrm{pos}(i,c')$ implies $\mu(i,c) \leq \mu(i,c')$. Intuitively, $\mu(i,c)$ indicates how badly candidate c misrepresents voter i. A misrepresentation function is said to be *positional* if there exists a vector $\mathbf{s} = (s_1, \ldots, s_m) \in \mathbb{Z}_+^m$ such that $\mu(i,c) = s_{\mathrm{pos}(i,c)}$; note that this implies $s_1 \leq s_2 \leq \cdots \leq s_m$. We will refer to the vector $\mathbf{s}$ as the *misrepresentation vector* of μ. Throughout the paper we assume that $s_1 = 0$, i.e., each voter is perfectly represented by his favorite candidate. We will refer to the positional misrepresentation function that corresponds to the vector $(0, 1, \ldots, m-1)$ as the *Borda misrepresentation function* and denote it by μ_B.

Given an election $E = (C, V)$, a set of candidates $C' \subseteq C$, and a misrepresentation function $\mu : V \times C \to \mathbb{Z}_+$, we set

$$m_\mu^+(E, C') = \sum_{i \in V} \mu(i, \mathrm{top}(i, C'))$$

and

$$m_\mu^{\max}(E, C') = \max_{i \in V} \mu(i, \mathrm{top}(i, C')).$$

We are now ready to define the two computational problems that will be considered in this paper.

Definition 2.1. *An instance of* CC$^+$-MULTIWINNER *(respectively,* CC$^{\max}$-MULTIWINNER*) problem is given by an election $E = (C, V)$, an assembly size $k \in \mathbb{Z}_+$, a misrepresentation function $\mu : V \times C \to \mathbb{Z}_+$, and a bound B. It is a "yes"-instance if there is a subset of candidates $C' \subseteq C$ with $|C'| = k$ such that $m_\mu^+(E, C') \leq B$ (respectively, $m_\mu^{\max}(E, C') \leq B$) and a "no"-instance otherwise*[1].

We will sometimes consider the complexity of these problems for specific families of misrepresentation functions. Note that a misrepresentation function is defined for fixed C and V, so the question of asymptotic complexity makes sense for families of misrepresentation functions (parameterized by (C, V)), but not for individual misrepresentation functions. For instance, the Borda misrepresentation function can be viewed as a family of misrepresentation functions, as it is well-defined for any C and V, and in Section 5.1 we discuss the complexity of CC$^+$-MULTIWINNER with the additional constraint that μ is the Borda misrepresentation function.

Preferences That Are Single-Peaked on a Tree Recall that a *tree* is a connected graph that has no cycles; a *leaf* of a tree is a vertex of degree 1. A *path* is a tree that has exactly two leaves. A *star* is a tree that has exactly one vertex that is not a leaf; this vertex is called the *center* of the star. The *diameter* of a tree T is the maximum distance between two vertices of T; e.g., the diameter of a star is 2. Another important combinatorial parameter of a tree is its *pathwidth*. We omit the formal definition of this notion (see [Robertson and Seymour, 1983]); intuitively, the pathwidth of T is a measure of how close T is to being a path. Given a parameter λ, we will say that a tree T is *λ-narrow* if it has at most λ leaves.

Consider a tree T with a vertex set C. An election $E = (C, V)$ is said to be *single-peaked on* T [Demange, 1982] if for every voter $i \in V$, every pair of candidates $c, c' \in C$, and every candidate c'' that lies on the (unique) c–c' path in T it holds that $\mathrm{pos}(i, c'') < \max\{\mathrm{pos}(i, c), \mathrm{pos}(i, c')\}$. An election $E = (C, V)$ is said to be *single-peaked on a tree* if there exists a tree T with the vertex set C such that E is single-peaked on T; E is said to be *single-peaked* if it is single-peaked on some tree T that is a path. There exist efficient algorithms for determining whether a given election is single-peaked on a tree [Trick, 1989] or single-peaked [Bartholdi and Trick, 1986; Escoffier *et al.*, 2008]; when the answer is positive, these algorithms explicitly construct a tree (respectively, a path) that witnesses this.

3 CC$^{\max}$-Multiwinner on Arbitrary Trees

We start by presenting a greedy algorithm for CC$^{\max}$-MULTIWINNER that works on arbitrary trees. Our algorithm proceeds by finding an assembly of minimum size that satisfies a given worst-case misrepresentation bound.

First, observe that CC$^{\max}$-MULTIWINNER can be reduced to the following variant of the HITTING SET problem, where the ground set is the node set of a tree, and we need to hit a collection of subtrees.

Definition 3.1. *An instance of* TREE HITTING SET *problem is given by a tree T on node set C, a set $\mathcal{T} = \{T_1, \ldots, T_n\}$ of subtrees of T and a target cover size $k \in \mathbb{Z}_+$. It is a "yes"-instance if there is a subset of nodes $C' \subseteq C$ with $|C'| \leq k$ such that $C' \cap T_i \neq \emptyset$ for $i = 1, \ldots, n$ and a "no"-instance otherwise.*

Given an instance of CC$^{\max}$-MULTIWINNER, we construct a TREE HITTING SET instance as follows: The ground set is the candidate set C, the tree T is the tree with respect to which voters' preferences are single-peaked, and the target cover size equals the assembly size k. For each $i \in \{1, \ldots, n\}$ the subtree T_i is built from voter i's preferences: it is the induced subtree on $C_i^B = \{c \in C \mid \mu(i,c) \leq B\}$. Since μ is monotone, the set C_i^B is a prefix block of i's preference order, i.e., is of the form $\{c \in C \mid \mathrm{pos}(i,c) \leq b\}$ for some $b \in \{1, \ldots, m\}$, and therefore (see Theorem 1 in [Trick, 1989]) it induces a subtree of T. The correctness of the reduction follows from the fact that $m_\mu^{\max}(E, C') \leq B$ if and only if $C' \cap \{c \in C \mid \mu(i,c) \leq B\} \neq \emptyset$ for all i.

For TREE HITTING SET we give a greedy algorithm (Algorithm 1). In the description of the algorithm, d is the distance in T. Note that by our choice of r_i in the **while** loop we have $r_i \in T_j$ for any tree $T_j \in \mathcal{T}$ such that $T_i \cap T_j \neq \emptyset$. This observation is crucial for the proof of correctness; the formal proof proceeds by induction on the size of the tree.

[1] Under our definition it may happen that some candidate in the assembly does not represent any voter, i.e., there exists a $c' \in C'$ such that $c' \neq \mathrm{top}(i, C')$ for all $i \in V$; equivalently, we allow for assemblies of size $k' < k$. It is assumed that the weight of such candidate in the resulting assembly will be 0. This definition is also used, by e.g., [Cornaz *et al.*, 2012; Skowron *et al.*, 2013]. In contrast, Betzler *et al.* [2013] define the Chamberlin–Courant rule by explicitly specifying an assignment of voters to candidates, so that each candidate in C' has at least one voter who is assigned to it. The resulting voting rule is somewhat harder to analyze algorithmically. Note that when $|\{\mathrm{top}(i, C) \mid i \in V\}| \geq k$, the two variants of the Chamberlin–Courant rule coincide.

Algorithm 1: TREE HITTING SET$(T, \mathcal{T}, k)$

Pick some node r^* as the root of T;
for *each* $T_i \in \mathcal{T}$ **do**
  pick $r_i \in T_i$ so that $d(r^*, r_i) \leq d(r^*, r)$ for all $r \in T_i$;
$R \leftarrow \{r_1, \ldots, r_n\}$, $H \leftarrow \emptyset$;
while $R \neq \emptyset$ **do**
  Pick some r_i in $\arg\max_{r_j \in R} d(r^*, r_j)$;
  $R' \leftarrow \{r_j \in R \mid r_i \in T_j\}$;
  $H \leftarrow H \cup \{r_i\}$, $R \leftarrow R \setminus R'$;
if $|H| \leq k$ **then**
  return "yes";
else
  return "no";

Theorem 3.2. *The problem* $\text{CC}^{\max}$-MULTIWINNER *is polynomial-time solvable for elections that are single-peaked on a tree.*

4 CC$^+$-Multiwinner on Narrow Trees

For CC$^+$-MULTIWINNER we have an algorithm whose running time is polynomial for any election that is single-peaked on a tree with a constant number of leaves.

Theorem 4.1. *Given an election $E = (C, V)$ with $|C| = m$, $|V| = n$ and a tree T with λ leaves such that E is single-peaked on T, we can solve* CC$^+$-MULTIWINNER *in time* $\text{poly}(n, m^\lambda, k^\lambda)$*, where k is the target assembly size.*

Proof. We use dynamic programming to find an assembly of size at most k that minimizes the total misrepresentation.

We pick an arbitrary node r^* to be the root of T. This choice induces a partial order $\succ$ on C: we set $a \succ b$ if a lies on the (unique) path from r^* to b in T. A set $A \subseteq C$ is said to be an *anti-chain* if no two elements of A are comparable with respect to $\succ$. Observe that for every assembly $S \subseteq C$, its set of maximal elements with respect to $\succ$ forms an anti-chain. Note also that if a and b belong to an anti-chain $A \subseteq C$ and c is a leaf of T, then it cannot be the case that both a and b are ancestors of c, and hence $|A| \leq \lambda$.

Given a node r, let T_r be the subtree of T rooted at r, i.e., the node set of T_r is $C_r = \{r\} \cup \{c \mid r \succ c\}$. Let V_r be the set of all voters whose most preferred candidate belongs to C_r, and let E_r be the election obtained from E by restricting the candidate set to C_r and the voter set to V_r. For each $r \in C$ and each $\ell = 1, \ldots, k$ let $M(r, \ell)$ be the smallest total misrepresentation for E_r that can be achieved by an assembly of size at most ℓ (using candidates in C_r only), subject to r being selected. Suppose that we have computed these quantities for all descendants of r; we will now explain how to compute them for r.

Let S be an optimal assembly of size at most ℓ for E_r. Let $A = \{r_1, \ldots, r_s\}$ be the set of maximal elements of $S \setminus \{r\}$ with respect to $\succ$ and let $\ell_j = |S \cap C_{r_j}|$ for $j = 1, \ldots, s$; we have $\ell_1 + \cdots + \ell_s = \ell - 1$. For each j the contribution of voters in V_{r_j} to the total misrepresentation is given by $M(r_j, \ell_j)$: any such voter is better represented by r_j than by any candidate not in C_{r_j}. On the other hand, consider a voter i in $V_r \setminus (V_{r_1} \cup \cdots \cup V_{r_s})$. His most preferred candidate in S is one of $r, r_1, \ldots, r_s$: for each $j = 1, \ldots, s$, candidate r_j is a better representative for i than any other candidate in C_{r_j}.

This suggests the following procedure for computing $M(r, \ell)$. Let $\mathcal{T}_r$ be the set of all anti-chains in T_r. An *ℓ-division scheme* for an anti-chain $A = \{r_1, \ldots, r_s\} \in \mathcal{T}_r$ is a list $L = (\ell_1, \ldots, \ell_s)$ such that $\ell_j \geq 1$ for all $j = 1, \ldots, s$ and $\ell_1 + \cdots + \ell_s = \ell$; we denote by $\mathcal{L}_\ell^A$ the set of all ℓ-division schemes for A. For every $A = \{r_1, \ldots, r_s\} \in \mathcal{T}_r \setminus \{\{r\}\}$, every $L = (\ell_1, \ldots, \ell_s) \in \mathcal{L}_{\ell-1}^A$, set $V_r' = V_r \setminus (V_{r_1} \cup \cdots \cup V_{r_s})$ and

$$M(A, L) = \sum_{j=1}^{s} M(r_j, \ell_j) + \sum_{i \in V_r'} \mu(i, \text{top}(i, A \cup \{r\})).$$

We then have $M(r, \ell) = \min_{A \in \mathcal{T}_r \setminus \{\{r\}\}, L \in \mathcal{L}_{\ell-1}^A} M(A, L)$. The base case for this recurrence corresponds to the case when r is a leaf, and is easy to deal with.

The final answer depends on whether the root r^* is selected; if not, we minimize over all antichains $A = \{r_1, \ldots, r_s\} \in \mathcal{T}_r \setminus \{\{r\}\}$ and over all ways of dividing the k slots $L = (\ell_1, \ldots, \ell_s) \in \mathcal{L}_k^A$. That is, we set

$$M'(A, L) = \sum_{j=1}^{s} M(r_j, \ell_j) + \sum_{i \in V_r'} \mu(i, \text{top}(i, A)).$$

Note that here r^* does not appear in the second term. The total misrepresentation is given by $\min\{M(r^*, k), \min_{A \in \mathcal{T}_r \setminus \{\{r\}\}, L \in \mathcal{L}_k^A} M'(A, L)\}$.

We have argued that the size of each anti-chain is at most λ. Therefore, at each node we enumerate at most m^λ anti-chains and at most k^λ divisions. This establishes our bound on the running time. □

5 Hardness of CC$^+$-Multiwinner on Arbitrary Trees

The algorithm presented in Section 4 is only guaranteed to run in polynomial time on λ-narrow trees, where λ is bounded by a constant. This is somewhat disappointing, as many problems on graphs can be solved efficiently on arbitrary trees or at least on bounded-degree trees. Thus, it is natural to ask if single-peakedness on an arbitrary tree is, in fact, sufficient for the existence of a polynomial-time algorithm for CC$^+$-MULTIWINNER. In this section, we will show that this is unlikely to be the case, at least for arbitrary (positional) misrepresentation functions. Specifically, we will show that CC$^+$-MULTIWINNER remains NP-hard even if the voters' preferences are single-peaked on a star (which has bounded diameter and pathwidth). We will then modify our NP-hardness reduction to show that our problem is also hard on the caterpillar graph (which has bounded degree). Our hardness results hold even if μ is required to be a positional misrepresentation function that can take at most three distinct values, i.e., there exist some s, S, $0 < s < S$, such that $\mu(i, c) \in \{0, s, S\}$ for all $i \in V$, $c \in C$.

We will first characterize the elections that are single-peaked on a star; this characterization is implicit in [Demange, 1982]. We omit the proof due to space constraints.

Proposition 5.1. *An election $E = (C, V)$ is single-peaked on a star if and only if there exists a candidate $a \in C$ such that $\mathrm{pos}(i, a) \in \{1, 2\}$ for every $i \in V$.*

We are now ready to present our first hardness result.

Theorem 5.2. CC$^+$-MULTIWINNER *is* NP*-hard even for elections that are single-peaked on a star. The hardness result holds for any family of positional misrepresentation functions whose misrepresentation vectors* $\mathbf{s}$ *satisfy $s_1 = 0$, $s_2 = \cdots = s_\ell > 0$, $s_{\ell+1} > s_\ell$ for some $\ell \geq 5$.*

Proof. We will reduce a restricted version of EXACT COVER BY 3-SETS (X3C) to our problem. Recall that an instance of X3C is given by a ground set $X = \{x_1, \ldots, x_p\}$ with $p = 3p'$ for some $p' \in \mathbb{Z}_+$ and a collection $\mathcal{Y} = \{Y_1, \ldots, Y_q\}$ of 3-element subsets of X; it is a "yes"-instance if we can pick a subcollection $\mathcal{Y}' \subseteq \mathcal{Y}$ of size p' that covers X, i.e., for each $x_i \in X$ there exists a Y_j in $\mathcal{Y}'$ such that $x_i \in Y_j$. This problem is known to be NP-hard even if each element of X appears in at most three sets in $\mathcal{Y}$ [Garey and Johnson, 1979].

Fix a family of positional misrepresentation functions μ that satisfy the condition in the statement of the theorem for some $\ell \geq 5$. Given an instance $(X, \mathcal{Y})$ of X3C such that $|\{Y_j \in \mathcal{Y} \mid x_i \in Y_j\}| \leq 3$ for each $x_i \in X$, we construct an instance of our problem as follows. We set $Y = \{y_1, \ldots, y_q\}$, and let $C = \{a, z\} \cup X \cup Y \cup D$, where $D = \{d_{i,j}\}_{i=1,\ldots,p}^{j=1,\ldots,\ell}$ is the set of dummy candidates. For each $Y_j \in \mathcal{Y}$ we construct $p + 1$ voters who rank y_j first, a second and z third, followed by all other candidates in an arbitrary order; let V_1 denote the set of voters constructed in this way. Further, for each $x_i \in X$, we construct a voter who ranks x_i first, followed by a, followed by the candidates y_j such that $x_i \in Y_j$ (in an arbitrary order), followed by $d_{i,1}, \ldots, d_{i,\ell}$, followed by all other candidates in an arbitrary order. Denote the resulting set of voters by V_2. Finally, let V_3 be a set of $(p + 1)(q + 1)$ voters who all rank z first and a second, followed by all other candidates in an arbitrary order. Let $V = V_1 \cup V_2 \cup V_3$, and set $E = (C, V)$. Also, set $B = s_2((p + 1)(q - p') + p)$ and $k = p' + 1$. This completes the description of our instance of the CC$^+$-MULTIWINNER problem. Observe that every voter in V ranks a second, so by Proposition 5.1 the election E in single-peaked on a star. Due to space constraints, we omit the proof that our reduction is correct. □

The construction used in the proof of Theorem 5.2 can be modified to show that CC$^+$-MULTIWINNER remains NP-hard on trees of maximum degree 3; the basic idea is to "clone" candidate a so that the resulting election becomes single-peaked on a caterpillar graph.

Corollary 5.3. CC$^+$-MULTIWINNER *is* NP*-hard even for elections that are single-peaked on trees of maximum degree* 3*.*

5.1 CC$^+$-Multiwinner Under Borda Misrepresentation Function on Stars

Observe that the Borda misrepresentation function does not satisfy the conditions of Theorem 5.2. In fact, we will now show that, somewhat surprisingly, for elections that are single-peaked on a star, CC$^+$-MULTIWINNER under the Borda misrepresentation function is polynomial-time solvable. It is an open question whether this remains to be the case for elections that are single-peaked on arbitrary trees; we conjecture that the answer to this question is negative.

Proposition 5.4. *The problem* CC$^+$-MULTIWINNER *under the Borda misrepresentation function is polynomial-time solvable for elections that are single-peaked on a star.*

Proof. Consider an election $E = (C, V)$ that is single-peaked on a star, and let a be the center of this star. By Proposition 5.1, every voter in V ranks a first or second. For each $c \in C$, let $V_c = \{i \in V \mid \mathrm{top}(i, C) = c\}$.

Let k be the target assembly size. We claim that the following greedy algorithm finds an optimal assembly of size k (we omit the proof due to space constraints). First, we sort the candidates in $C \setminus \{a\}$ according to the number of times they are ranked first by the voters (i.e., so that if b precedes c in the sorted order then $|V_b| \geq |V_c|$), breaking ties arbitrarily. We then construct two assemblies as follows. The first assembly consists of a and top $k - 1$ candidates in the sorted order. The second assembly consists of the top k candidates in the sorted order. We then compute the misrepresentation for both assemblies, and pick the better of the two. □

The proof of Proposition 5.4 goes through for every positional misrepresentation function whose misrepresentation vector $\mathbf{s}$ satisfies $s_1 = 0$, $s_3 \geq 2s_2 > 0$. Further, if $s_2 = 0$ then any assembly that contains the center of the star is clearly optimal. We obtain the following corollary.

Corollary 5.5. *The problem* CC$^+$-MULTIWINNER *is polynomial-time solvable for elections that are single-peaked on a star under every positional misrepresentation function whose misrepresentation vector* $\mathbf{s}$ *satisfies $s_1 = 0$, $s_3 \geq 2s_2$.*

6 Algorithm for Building Narrow Trees

In this section, we present a polynomial-time algorithm that given an election E outputs a tree T with minimum number of leaves among all trees on which E is single-peaked; it returns failure if E is not single-peaked on a tree.

Our algorithm is an adaptation of that of Trick's [1989]. Trick's algorithm builds the tree T recursively: it identifies a set L of candidates that have to be leaves, attaches each leaf $c \in L$ to an arbitrary candidate in its allowed set of neighbors R_c, and then recurses on the subset of candidates with L eliminated. In more detail, L is the set of candidates that are least preferred by some voter: If the current candidate set is C' then $L = \{\mathrm{bot}(i, C') \mid i \in V\}$. The set R_c is defined as $R_c = \cap_{i \in V} \mathrm{pre}(c, i, C')$, where $\mathrm{pre}(c, i, C')$ is c's allowed set of neighbors with respect to $\succ_i$: $\mathrm{pre}(c, i, C') = \{\mathrm{top}(i, C' \setminus \{c\})\}$ if $c = \mathrm{top}(i, C')$ and $\mathrm{pre}(c, i, C') = \{c' \in C' \mid c' \succ_i c\}$ otherwise. For $|C'| \geq 3$, R_c is a subset of $C' \setminus L$, so the algorithm does build a tree. The correctness of the algorithm is based on the fact that in any tree realizing single-peakedness a leaf $c \in L$ has to be adjacent to some $c' \in R_c$, but there are no further constraints on choosing c's neighbor. No tree exists if and only if some $R_c = \emptyset$. More details can be found in [Trick, 1989].

Algorithm 2: BUILD-NT($E = (C, V)$)

```
Construct an empty bipartite graph G where both parts A
and B are equal to the candidate set C;
C' ← C;
while |C'| ≥ 3 do
    L = {bot(i, C') | i ∈ V};
    for each candidate c ∈ L do
        R_c = ∩_{i∈V} pre(c, i, C');
        if R_c = ∅ then
            return failure;
        else
            Connect c in part A to each c' ∈ R_c in part
            B;
    C' ← C' \ L;
if |C'| = 2 then
    Add to T the edge between the two candidates in C';
Find a maximum matching M in bipartite graph G;
for each c ∈ A \ C' do
    if c is matched in M to candidate c' then
        Add edge (c, c') to T;
    else
        Add edge (c, c') to T where c' is any neighbor of
        c in G;
return T;
```

Our algorithm essentially adopts the same structure, but defers the attachment of leaves till the very end. Along the way, we construct a bipartite graph G storing all information on R_c. It turns out that a maximum matching of G corresponds to an optimal way to attach all leaves identified in different rounds of the **while** loop.

Theorem 6.1. *Given an election E, Algorithm 2 outputs a tree T if and only if E is single-peaked on a tree. Moreover, T has the smallest number of leaves among all trees witnessing that E is single-peaked on a tree. The running time of Algorithm 2 is polynomial in the input size.*

Proof. Compared to Trick's algorithm, our algorithm is only more careful about which candidate in R_c to pick when attaching each leaf c, so when it outputs a tree T, the input election is guaranteed to be single-peaked on T. It remains to show that T has as few leaves as possible.

To see this, suppose that the body of the **while** loop is executed $f-1$ times, let L_j be the set of leaves identified in the j-th iteration, $j \leq f-1$, and let $L_f = C \setminus \cup_{j=1,\ldots,f-1} L_j$. For every $k \leq f-1$ and every $c \in L_k$ we have $R_c \subseteq \cup_{j>k} L_j$.

We have $|L_f| \leq 2$ and $|L_{f-1} \cup L_f| \geq 3$ by the condition of the **while** loop. Note that $L_f \neq \emptyset$ if the input election is single-peaked on a tree: otherwise, every $c \in L_{f-1}$ would have $R_c \subseteq L_f = \emptyset$, a contradiction with $R_c \neq \emptyset$. Hence $|L_f| \in \{1, 2\}$. Further, $|L_f| = 1$ implies $|L_{f-1}| \geq 2$.

The set sequence $L_1, \ldots, L_f$ is fully determined by the input; to build a tree realizing single-peakedness, the only freedom we have is to decide on the leaf attachment, which can be any mapping $H : C \setminus L_f \to C$ such that $H(c) \in R_c$ for all $c \in C \setminus L_f$. Our goal is to pick a mapping H that minimizes the number of leaves of the output tree.

First, we observe that $H(C \setminus L_f)$ is exactly the set of internal nodes of the output tree. Thus, minimizing the number of leaves is equivalent to maximizing $|H(C \setminus L_f)|$. To see this, note that $c \in H(C \setminus L_f)$ for some $c \in L_k$ implies that there is a node $c' \in L_{k'}$ with $k' < k$ that is attached to c. If $k < f$, c is also connected to some $c'' = H(c) \in L_{k''}$ with $k < k''$. Since $k' \neq k''$ and $c' \neq c''$, the degree of node c is at least 2. If $c \in L_f$ and $|L_f| = 2$, c is connected to the other candidate in L_f, thus also has degree at least 2. Finally, if $L_f = \{c\}$, then $|L_{f-1}| \geq 2$, and each candidate in L_{f-1} can only be attached to c. This shows that $H(C \setminus L_f)$ consists of internal nodes only. On the other hand, every candidate $c \notin H(C \setminus L_f)$ has degree 1.

Now it remains to show that the leaf attachment in our algorithm maximizes $|H(C \setminus L_f)|$. It is easy to verify that our extension of a maximum matching M of G (attach every unattached leaf to any neighbor) produces a legal leaf attachment mapping (with part A as domain and part B as image), which we denote by H_0. The range of H_0 is exactly the set of matched candidates in B, otherwise a matching of larger size exists. Thus $|H_0(C \setminus L_f)| = |M|$. Now, for the sake of contradiction, suppose there is a leaf attachment mapping H' with a larger range. Then, by choosing an arbitrary preimage for each $c \in H'(C \setminus L_f)$, we obtain a matching M' with $|M'| = |H'(C \setminus L_f)| > |H_0(C \setminus L_f)| = |M|$, contradicting the maximality of M.

Therefore, our algorithm produces a leaf attachment mapping with maximum range size, and minimizes the number of leaves in the output tree. It runs in polynomial time since a maximum matching can be efficiently computed. □

7 Conclusions and Future Work

We have studied the complexity of winner determination under Chamberlin–Courant's rule for elections that are single-peaked on a tree. For the egalitarian variant of this rule, we showed that this problem is polynomial-time solvable for any tree and any misrepresentation function. For the utilitarian setting, our algorithm runs in polynomial time when the tree has a small number of leaves, and our hardness results show that we cannot replace the constraint on the number of leaves with one on the pathwidth, diameter, or the maximum degree. However, intriguingly, our hardness results do not apply to the Borda misrepresentation function μ_B, and we have presented evidence that restricting attention to μ_B may simplify our problem. Thus, the most obvious open question suggested by our work is whether it is easy to determine the Chamberlin–Courant's winners when voters' preferences are single-peaked on a tree and the misrepresentation function is given by μ_B. It would also be interesting to see whether our easiness results for preferences that are single-peaked on a tree extend to the egalitarian version of Monroe's rule (winner determination under the utilitarian version of this rule is known to be hard even when preferences are single-peaked on a path [Betzler *et al.*, 2013]).

Acknowledgments This research was supported by National

Research Foundation (Singapore) under grant RF2009-08 (Edith Elkind, Lan Yu), and by EAPSI and National Science Foundation (USA) under grant OISE-1209805 (Hau Chan).

References

[Arrow *et al.*, 2002] K. Arrow, A. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare, Volume 1*. Elsevier, 2002.

[Bartholdi and Trick, 1986] J. Bartholdi and M. Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(4):165–169, 1986.

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

[Betzler *et al.*, 2013] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *Journal of AI Research*, 2013. to appear.

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

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

[Cornaz *et al.*, 2012] D. Cornaz, L. Galand, and O. Spanjaard. Bounded single-peaked width and proportional representation. In *ECAI'12*, pages 270–275, 2012.

[Demange, 1982] G. Demange. Single-peaked orders on a tree. *Mathematical Social Sciences*, 3(4):389–396, 1982.

[Escoffier *et al.*, 2008] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *ECAI'08*, pages 366–370, 2008.

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

[Hemaspaandra *et al.*, 1997] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of ACM*, 44(6):806–825, 1997.

[Hemaspaandra *et al.*, 2005] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

[Lu and Boutilier, 2011] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *IJCAI'11*, pages 280–286, 2011.

[Monroe, 1995] B. Monroe. Fully proportional representation. *American Political Science Review*, 89(4):925–940, 1995.

[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.

[Robertson and Seymour, 1983] N. Robertson and P. Seymour. Graph minors. I. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983.

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

[Skowron *et al.*, 2013] P. Skowron, P. Faliszewski, and A. Slinko. Proportional representation as resource allocation: Approximability results. In *IJCAI'13*, 2013.

[Trick, 1989] M. Trick. Recognizing single-peaked preferences on a tree. *Mathematical Social Sciences*, 17(3):329–334, 1989.