# Parameterized Complexity of Winner Determination in Minimax Committee Elections

Hong Liu[*]
School of Computer Science and Technology,
Shandong University
1500 Shunhua Road, Jinan 250101, P.R. China
hong-liu@sdu.edu.cn

Jiong Guo[†]
School of Computer Science and Technology,
Shandong University
1500 Shunhua Road, Jinan 250101, P.R. China
jguo@sdu.edu.cn

## ABSTRACT

Recently, the winner determination problem of minimax approval committee (or referenda) elections has been extensively studied. In particular, Misra et al. [AAMAS 2015] investigated its parameterized complexity with respect to various parameters. Among others, they showed that this problem is FPT with the Hamming distance upper-bound as parameter. Following the suggestion of Baumeister and Dennisen [AAMAS 2015], we consider minimax committee election rules for trichotomous votes, linear orders, and partial orders. Complementing the classical complexity results for these rules by Baumeister et al. [ADT 2015], we show that the winner determination problem on trichotomous votes admits similar parameterized complexity behavior as on dichotomous votes. To this end, we fix a flaw in the parameterized algorithm given by Misra et al. [AAMAS 2015], which uses the Hamming distance upper-bound as parameter. Moreover, we prove that on both linear and partial orders, it is W[2]-hard to determine the winners with respect to the size of the committee. Even with both the size of the committee and the number of votes as parameters, this problem remains fixed-parameter intractable; here, it becomes W[1]-hard. We also present FPT algorithms for both linear and partial orders with the distance upper-bound as parameter. Finally, we show that in several special settings, the winner determination problem of a minimax approval committee election is polynomial-time solvable. These special settings have been considered for similar voting problems.

## Keywords

Computational Social Choice; Fixed-Parameter Tractability; Minimax Committee Election

## 1. INTRODUCTION

The problem of aggregating the preferences of different agents occurs in diverse situations and plays a fundamental role in artificial intelligence and social choice [9, 12]. Here, agents (voters) express their preferences over candidates and a voting rule is employed to determine which candidate wins. While the most studied setting is the single-winner case, voting can also be used to elect a fixed-size set of winners, a so-called committee. Real-world examples include parliamentary elections, selecting committees, understanding heredity and signals in DNA, and recommendation systems (see [3] and the references therein). Recently, we witness an increasing interest in the study of the axiomatic and algorithmic aspects of committee elections [1, 2, 3, 6, 11, 15, 20, 23, 22, 26]

**Approval voting.** A widely used rule for committee election is the approval voting (AV) approach, where the voters decide for each candidate, whether they approve or disapprove of this candidate. Such votes are called *approval votes*. The winning committee consists of the $k$ candidates with the most approvals. Brams et al. [8] suggested a distance-based variant of AV, called minimax approval voting, which selects a set of $k$ candidates that minimize the maximum Hamming distance from the given votes. Another popular distance-based AV variant is the minisum approval voting, whose optimization goal is to minimize the total Hamming distance of the committee to all given votes. Amanatidis et al. [1] introduced a so-called ordered weighted averaging operator to the distance-based AV approach, generalizing both minimax and minisum approval rules. Other variants of AV include proportional approval voting, reweighted approval voting, and satisfaction approval voting (see [3] for more details).

**Extensions of minimax and minisum AV.** In 2015, Baumeister and Dennisen [5] extended the minimax and minisum approaches to other forms of votes such as trichotomous votes, linear orders, and partial orders. A voting with trichotomous votes allows the voters in addition to approval and disapproval to abstain for a candidate [18], which could be of particular interest in the multiple referenda elections.[1] A trichotomous vote can be represented as a vector over the candidates, where 1 and -1 stand for approval and disapproval of candidates, and 0 stands for abstention. The committee is then a vector with entries from 1 and -1. Baumeister and Dennisen proposed a modified Hamming distance to measure the distance between the committee and the votes. In the case of linear orders, every voter's preference is represented as a total, transitive and asymmetric binary relation of the candidates and the rank-sum distance is used

[*]Partially supported by Natural Science Foundation of Shandong Province, China (No. ZR2013FM030)
[†]Partially supported by the Taishan project of Shandong, China

[1]Multiple referenda elections seek for a collective decision over several binary propositions and thus, share similar properties with committee elections.

**Appears in:** *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

to select the winning committee. Partial orders are also transitive and asymmetric, but not necessarily total. The dissatisfaction of the voters with a committee is measured by a generalized Kemeny distance. See the preliminaries for the definitions of the votes and distance functions. Baumeister et al. [6] proved that concerning the classical complexity, the minimax and minisum winner determination problems admit the same behavior for all three forms of votes. That is, as in the case of approval votes, the minisum version is polynomial-time solvable and the minimax version is NP-hard for all three forms of votes.[2]

**Parameterized Complexity.** We focus on the parameterized complexity of the above three extensions of the minimax approval voting. Parameterized complexity allows to give a more refined analysis of computational problems and in particular, can provide a deep exploration of the connection between the problem complexity and various problem-specific parameters [14]. In the last years, parameterized complexity analysis has been applied to diverse problems from computational social choice, including winner determination, control, and bribery [7, 10]. Misra et al. [24] initialized the study of parameterized complexity of the minimax approval voting. They showed, among others, that the winner determination problem of minimax approval voting is fixed-parameter tractable with respect to the Hamming distance upper-bound, the number of candidates, or the number of candidates, while it becomes fixed-parameter intractable (W[2]-hard) with respect to the size of the committee.

**Our results.** As in [24], we also consider four parameters: the number $n$ of votes, the number $m$ of candidates, the size $k$ of committee, and the distance upper-bound $d$.

Our main findings is that, in contrast to the classical complexity, the winner determination problem admits different parameterized complexity behavior for the considered vote forms. First, compared to the case with approval votes, the introduction of trichotomous votes essentially has no impact on the parameterized complexity with respect to all four parameters. Hereby, we identify and fix a flaw in the algorithm by Misra et al. for the case of approval votes with $d$ as parameter. Moreover, we propose two parameterized algorithms for the linear and partial orders cases with $d$ as parameter, which are based on completely different idea than the one for the approval and trichotomous votes cases and need much more algorithmic effort. However, the resulting running times are comparable with the ones for the approval and trichotomous votes cases. In particular, we show that with the linear and partial orders, the winner determination problem becomes fixed-parameter intractable, i.e., W[1]-hard, even with both $k$ and $n$ as parameters, in contrast to the FPT-result with the same parameterization for the approval and trichotomous votes cases. Finally, we present polynomial time algorithms for two special cases of the maximin approval voting with approval votes. These two special cases have been studied for similar voting problems [16, 17].

## 2. PRELIMINARIES

In this section, we introduce the definitions and notations used in this paper. An *election* is a pair $(C, V)$, where $C = \{c_1, \ldots, c_m\}$ is a set of candidates (or alternatives) and $V = \{v_1, \ldots, v_n\}$ is an ordered list of voters. Each voter's preference among the candidates in $C$ is expressed through the vote he casts. We also use $V$ to denote the list of votes that the voters in $V$ cast. We will refer to the list $V$ as a *preference profile*, and denote the number of voters in $V$ by $n$, and the number of candidates in $C$ by $m$. A committee election is a triple $(C, V, k)$, where $k$ is the size of the committee. A *committee election system* is a mapping that takes as input an committee election $(C, V, k)$, and outputs a size-$k$ subset $K$ of $C$, i.e., the *committee*.

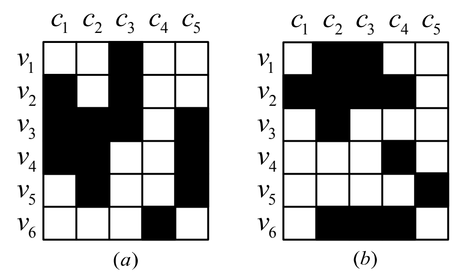

**Figure 1: Voter Interval and Candidate Interval. A black cell stands for a "1", and a white cell stands for a "0".**

As in [6], we consider four different forms of votes:

**Approval votes**. An approval vote $v$ on $C$ is a subset of $C$. While the corresponding voter approves of the candidates in $v$, he disapproves of the candidates in $C \setminus v$. For a fixed order of the candidates in $C$, each vote $v$ can also be represented as a $\{0,1\}^m$ vector, where a 1 stands for approval and a 0 for disapproval of the corresponding candidate. We use $v[i]$ to denote the $i$th entry of $v$. We call the number of 1's in an approval vote $v$ as its weight. For the same order of the candidates in $C$, a committee $K$ can also be represented by a $\{0,1\}^m$ vector with weight $k$.

The distance for an approval vote $v$ with a committee $K$ is defined as $HD(K, v) = \sum_{1 \le i \le m} |K[i] - v[i]| = |K \oplus v|$, that is, the Hamming distance between $K$ and $v$, where $K \oplus v$ is the symmetric difference of $K$ and $v$.

Given an election $(C, V)$, we say that $V$ satisfies the *voter interval* (VI) property if the votes in $V$ can be ordered such that for every candidate $c \in C$ the voters that approve of $c$ form an interval of that ordering [16]. See Figure 1$(a)$ for an example. We say that $V$ satisfies the *candidate interval* (CI) property if the candidates in $C$ can be ordered such that for every voter $v \in V$ the candidates that $v$ approves of form an interval of that ordering [17]. See Figure 1$(b)$ for an example.

**Trichotomous votes**. A trichotomous vote on $C$ is a partition of $C$ into three disjoint subsets $C_1$, $C_0$ and $C_{-1}$. The corresponding voter approves of the candidates in $C_1$, disapproves of the candidates in $C_{-1}$, and abstains for the candidates in $C_0$. For a fixed order of the candidates in $C$, each vote can also be represented as a $\{-1,0,1\}^m$ vector, where 1 stands for approval, $-1$ for disapproval and 0 for abstention. Given a trichotomous vote $v$ and a candidate $c$, we will denote the opinion of $v$ on $c$, i.e., 1, 0, or -1, by $value(v, c)$. We call the number of 1's in a trichotomous vote $v$ as its weight, and denote it by $|v|$. For the same order, a committee $K$ can be represented by a $\{1,-1\}^m$ vector with weight $k$.

[2] In [6], the complexity of the minimax version with linear orders was left open. According to [4], this problem is NP-hard as well.

The distance between two trichotomous votes $u, v$ is defined as $d_T(u,v) = \sum_{1\le i\le m} |u[i] - v[i]|$. The distance between a trichotomous vote $v$ and a committee $K$ is defined as $\delta(K, v) = d_T(K, v)$.

**Linear orders**. A linear order is a total, transitive, and asymmetric binary relation over $C$. We denote the vote of voter $v$ by $\succ_v$, where $c \succ_v d$ means that $v$ prefers candidate $c$ to candidate $d$. When the identity of the voter is clear from the context, we omit the subscript and write $\succ$ instead of $\succ_v$.

In this case, the distance for a vote $v$ with a committee $K$ is defined as the normalized sum of ranks of the committee members in vote $v$, i.e., $RS(K,v) = \sum_{c\in K} pos(c,v) - \frac{k(k+1)}{2}$, where $pos(c, v)$ denotes the position of candidate $c$ in vote $v$. For convenience, we can also express the distance in another way. Given a committee $K$, let $npos(c, v)$ denote the "normalized" position of candidate $c$ in vote $v$, i.e., the number of candidates that procede $c$ in $v$ but are not in $K$, or formally, $npos(c,v) = |\{c'|c' \in C \setminus K, c' \succ c\}|$. Then, $RS(K,v) = \sum_{c\in K} npos(c,v)$.

**Partial orders**. A partial order is a transitive and asymmetric, but not necessarily total binary relation over $C$. We denote the vote of voter $v$ also by $\succ_v$, where $c \succ_v d$ means that $v$ prefers candidate $c$ to candidate $d$. If the relation between $c$ and $d$ is unknown (or uncomparable), then we write $c \sim_v d$.

In this case, Baumeister and Dennisen [5] introduced a so-called "generalized Kemeny distance" between a committee $K$ and a vote $v$: $Dist(K,v) = \sum_{a,b\in C} d_{K,v}(a,b)$, where $d_{K,v}(a,b)$ is the distance between $K$ and $v$ regarding two candidates $a$ and $b$ defined as:

$$d_{K,v}(a,b) =$$

$$\begin{cases} 1 & \text{if } a \sim_v b \wedge ((a \in K \wedge b \notin K) \vee (a \notin K \wedge b \in K)), \\ 2 & \text{if } (a \in K \wedge b \notin K \wedge b \succ_v a) \vee \\ & \quad (a \notin K \wedge b \in K \wedge a \succ_v b), \\ 0 & \text{otherwise.} \end{cases}$$

We now define the social choice problem that is central to this work.

Definition 1. [$\Delta$-*Minimax-Voting*] *Given a list of votes $V = \{v_1, \ldots, v_n\}$ over a set of candidates $C = \{c_1, \ldots, c_m\}$, and two non-negative integers $d$ and $k$, find a size-$k$ subset $K$ of $C$ (committee) with $\max_{i=1,\ldots,n} \Delta(K, v_i) \le d$, where $\Delta \in \{HD, \delta, RS, Dist\}$ stands for the distance function for approval votes, trichotomous votes, linear orders or partial orders, respectively.*

Note that there might be more than one committee satisfying the distance upper-bound. For $\Delta$-Minimax-Voting, it suffices to find one of them.

Finally, we briefly introduce the relevant notions of parameterized complexity theory [13, 19, 25]. A parameterized decision problem is a language $\mathcal{L} \subseteq \Sigma^* \times \mathrm{N}$. $\mathcal{L}$ is *fixed-parameter tractable* (or in the class FPT) if for each $(x, k)$ in $\Sigma^* \times \mathrm{N}$, there exists an algorithm determining whether $(x, k)$ is in $\mathcal{L}$ with running time $O(f(k) \cdot |x|^c)$, where $f$ is some computable function, $c$ is a constant, and $k$ is the parameter. If no such algorithm exists for $\mathcal{L}$, $\mathcal{L}$ is called *fixed-parameter intractable*.

Given two parameterized problems $\mathcal{L}$ and $\mathcal{L}'$ (both encoded over $\Sigma^* \times \mathrm{N}$), we say $\mathcal{L}$ is *FPT-reducible* to $\mathcal{L}'$ if there is an algorithm that can map a given instance $(x, k)$ of $\mathcal{L}$ to an instance $(x', k')$ of $\mathcal{L}'$ in time $O(f(k) \cdot |x|^c)$, where $k' \le g(k)$, and $f$ and $g$ are computable functions, such that $(x, k)$ is a yes-instance of $\mathcal{L}$ if and only if $(x', k')$ is a yes-instance of $\mathcal{L}'$. Parameterized reductions are used to prove hardness of problems in the sense of parameterized complexity. The basic class of fixed-parameter intractable problems is W[1]. A parameterized problem $\mathcal{L}$ is said to be *$W[t]$-hard* for $t \ge 1$ if all the problems in $W[t]$ are FPT-reducible to $\mathcal{L}$.

The following table gives a summary of the parameterized complexity of the winner determination problem in the minimax committee elections for the above four forms of votes.

**Table 1: Summary of parameterized complexity results for $\Delta$-Minimax-Voting, where $\Delta \in \{HD, \delta, RS, Dist\}$ stands for the distance function for approval votes, trichotomous votes, linear orders or partial orders, respectively. The results with $m$ as parameter are trivial. New results of this paper are in boldface. W[1]-h. (or W[2]-h.) stands for W[1]-hard (or W[2]-hard).**

| | HD | $\delta$ | RS | Dist |
|---|---|---|---|---|
| $k$ | W[2]-h. [24] | W[2]-h. [24] | **W[2]-h.** | **W[2]-h.** |
| $d$ | FPT [24] | **FPT** | **FPT** | **FPT** |
| $n$ | FPT [24] | **FPT** | **W[1]-h.** | **W[1]-h.** |
| $m$ | FPT | FPT | FPT | FPT |
| $k, n$ | FPT [24] | **FPT** | **W[1]-h.** | **W[1]-h.** |

# 3. PARAMETERIZED COMPLEXITIES

## 3.1 $\delta$-Minimax-Voting

### 3.1.1 Parameter $d$: FPT

We show here that $\delta$-Minimax-Voting is fixed-parameter tractable with respect to the parameter $d$, by extending the algorithm by Misra et al. [24] for HD-Minimax-Voting.

The algorithm by Misra et al. starts with constructing a candidate committee. To do this, it modifies an arbitrary given vote $v$. If $v$ has exactly $k$ 1's, then $v$ is used as the candidate committee; if $v$ has more than $k$ 1's, the algorithm changes arbitrarily $|v| - k$ 1's to 0's. In the case of less than $k$ 1's, $k - |v|$ 0's are changed to 1's. Let $K$ denote the resulting vote. Then if there is a vote $v \in V$ with $HD(K, v) > 2d$, the algorithm outputs NOT FOUND; if there is no vote with $HD(K, v) > d$, then $K$ is output as solution. Otherwise, there exists at least one vote $v \in V$ that satisfies $d < HD(K, v) \le 2d$. For an arbitrary such vote, a search tree approach is applied. We note that this algorithm contains a flaw by giving a counterexample: $d = 3$, $k = 2$, $|C| = 8$, and $V = \{v_1 := 11111000, v_2 := 00011111\}$. If the algorithm uses $v_1$ to construct $K$ and changes the last three 1's to 0's, then we have $K = 11000000$ and $HD(K, v_2) > 2d$. This means that the algorithm reports NOT FOUND. However, it is not hard to check that there exists a unique solution 00011000 for the given instance. The problem of the algorithm lies in the newly constructed candidate committee $K$. Even if the distance between any pair of votes in the original $V$ is at most $2d$, the distance between $K$ and some vote in $V$ may be greater than $2d$. Thus, the algorithm by Misra et al. cannot always return the right answer.

By extending the upper-bound in the first if-condition from $2d$ to $3d$ and incorporating new idea dealing with trichotomous votes, we derive an algorithm for $\delta$-Minimax-Voting running in $O^*(d^d)$ time[3].

THEOREM 1. *$\delta$-Minimax-Voting can be solved in $O^*(d^d)$ time with the distance upper-bound $d$ as parameter and thus, is fixed-parameter tractable.*

### 3.1.2 Parameter $n$: FPT

THEOREM 2. *With $n$ as parameter, $\delta$-Minimax-Voting is fixed-parameter tractable.*

PROOF. We describe an integer linear program (ILP) with at most $2\cdot 3^n$ variables that solves $\delta$-Minimax-Voting. Fixed-parameter tractability then follows from the result that ILP is fixed-parameter tractable with the number of variables as parameter [21].

Consider the list of votes as an $n \times m$ matrix. Then each of the $m$ columns of this matrix is an element in $\{1, 0, -1\}^n$. Thus, there are at most $3^n$ different *column types*. Based on this observation, we can get an instance of ILP that is equivalent to the $\delta$-Minimax-Voting instance. Let $T$ denote the set of all column types, and for each type $t \in T$, let $n_t$ denote the number of columns in the input of type $t$. Also, let $\varphi_{t,i} \in \{1, 0, -1\}$ denote the letter (or value) at position $i$ in a given column type $t$.

The instance of ILP is defined over a set of $|\{1,-1\}|\cdot|T| \leq 2 \cdot 3^n$ variables, one variable $x_{t,\varphi}$ for each $t \in T$ and $\varphi \in \{1, -1\}$. The variable $x_{t,\varphi}$ contains the number of columns of type $t$ whose corresponding positions of the solution vector will be set to the value $\varphi$. The ILP instance then consists of the following constraints:

$$\sum_{t\in T}\sum_{\varphi\in\{1,-1\}} |\varphi - \varphi_{t,i}| \cdot x_{t,\varphi} \leq d \qquad \forall 1 \leq i \leq n$$

$$\sum_{\varphi\in\{1,-1\}} x_{t,\varphi} = n_t \qquad \forall t \in T$$

$$\sum_{t\in T} x_{t,1} = k$$

$$x_{t,\varphi} \in \{0, 1, 2, \dots\} \qquad \forall t \in T, \forall \varphi \in \{1, -1\}$$

Let $K$ denote a vector formed from a feasible solution to the above ILP. The $i$-th constraint of the first type ensures that $d_T(K, v_i) \leq d$ for vote $v_i$. It is not difficult to verify that the above ILP has a feasible solution if and only if the $\delta$-Minimax-Voting instance has a solution. $\square$

## 3.2 RS-Minimax-Voting

### 3.2.1 Parameter $k$: W[2]-Hard

The W[2]-hardness result is proved by a reduction from the W[2]-complete problem Dominating Set [13]. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$, the *closed neighborhood* of a vertex $v_i \in \mathcal{V}$ is defined to be $\{v_j | \{v_i, v_j\} \in \mathcal{E}\} \cup \{v_i\}$, denoted by $\mathbb{N}[v_i]$. A subset $\mathcal{V}' \subseteq \mathcal{V}$, where $\mathcal{V}' = \{v'_1, v'_2, \cdots, v'_{k'}\}$, is called a *size-$k'$ dominating set* of $\mathcal{G}$ if and only if every vertex $v \in \mathcal{V}$ is adjacent to at least one vertex in $\mathcal{V}'$, i.e., $v$ is in at least one of the closed neighborhoods $\mathbb{N}[v'_1], \dots, \mathbb{N}[v'_{k'}]$. Dominating Set asks for a size-$k'$ dominating set.

[3]In the $O^*$ notation we omit the polynomial terms, so that $O^*(f(d))$ stands for $O(f(d)p(n))$ for some polynomial $p$.

THEOREM 3. *RS-Minimax-Voting is $W[2]$-hard with respect to the parameter $k$.*

PROOF. Given a DOMINATING SET instance $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), k')$, where $\mathcal{V} = \{v_1, \dots, v_n\}$, we construct a RS-MINIMAX-VOTING instance $(A, P, d, k)$ as follows:

Let $k = k'$ and $d = (\alpha + 2n)(k - 1) + n$, where $\alpha = n(k-1) + 1$. With every $v_i \in \mathcal{V}$, we associate a candidate $c_i$. Denote by $C$ the set $\{c_1, \dots, c_n\}$. Additionally, for each $v_i \in \mathcal{V}$, we introduce three sets of dummy candidates, $X_i = \{x_{i1}, \dots, x_{in}\}$, $Y_i = \{y_{i1}, \dots, y_{i\alpha}\}$, and $Z_i = \{z_{i1}, \dots, z_{i\beta}\}$, where $\beta = k(\alpha + 2n)$. Thus, $A = C \cup X_1 \cup \dots \cup X_n \cup Y_1 \cup \dots \cup Y_n \cup Z_1 \cup \dots \cup Z_n$. For every vertex $v_i \in \mathcal{V}$, we have a corresponding vote $v_i$. The set of candidates corresponding to the vertices in $\mathbb{N}[v_i]$ is denoted by $\mathbb{C}[c_i]$, i.e., $\mathbb{C}[c_i] = \{c_j | v_j \in \mathbb{N}[v_i]\}$. For every $v_i$, we arbitrarily select $n - |\mathbb{C}[c_i]|$ elements from $X_i$ and put them in a set denoted by $X'_i$. The vote $v_i$ is set as follows:

$$\mathbb{C}[c_i] \succ X'_i \succ Y_i \succ C \setminus \mathbb{C}[c_i] \succ X_i \setminus X'_i \succ Z_i \succ \bigcup_{j\in[n], j\neq i} X_j \succ \bigcup_{j\in[n], j\neq i} Y_j \succ \bigcup_{j\in[n], j\neq i} Z_j,$$

where $R \succ S$ for two sets of candidates $R$ and $S$ means that $\forall a \in R$ and $b \in S$, we have $a \succ b$ but the ordering of $R$ (or $S$) can be arbitrary.

In the following, we show that the Dominating Set instance has a size-$k$ dominating set if and only if there is a size-$k$ subset $K$ of $A$ such that $\max_{i=1,\dots,n} RS(K, v_i) \leq d$.

$\Rightarrow$: Suppose that there exists a dominating set $\mathcal{V}' = \{v'_1, \dots, v'_k\}$ in $\mathcal{G}$. Denote by $K$ the set of candidates $c'_1, \dots, c'_k$ in $C$ that correspond to the vertices in $\mathcal{V}'$. For each $v_i \in P$, at least one candidate $c'_i \in K$ $(1 \leq i \leq k)$ is in $\mathbb{C}[c_i]$ and $npos(c'_i, v_i) < n$. On the other hand, we have $|K \cap (C \setminus \mathbb{C}[c_i])| \leq k - 1$ and, for each candidate in $K \cap (C \setminus \mathbb{C}[c_i])$, its "normalized" position in $v_i$ is less than $\alpha + 2n$. It is easy to see that $RS(K, v_i) < n + (\alpha + 2n)(k - 1) = d$. Thus, $\max_{i=1,\dots,n} RS(K, v_i) \leq d$.

$\Leftarrow$: Now suppose there is a size-$k$ subset $K = \{c'_1, \dots, c'_k\}$ of $A$ such that $\max_{i=1,\dots,n} RS(K, v_i) \leq d$. Denote by $\mathcal{V}'$ the set of vertices $v'_1, \dots, v'_k$ in $\mathcal{V}$ that correspond to the candidates in $K$.

First, we claim that, in each vote $v_i$, none of the candidates in $K$ appears after $Z_i$. If this is not true, there is a candidate $c \in K$ that lies after $Z_i$ in vote $v_i$. Then, $npos(c, v_i) \geq |C| + |X_i| + |Y_i| + |Z_i| - \frac{k(k-1)}{2} = 2n + \alpha + k(\alpha + 2n) - \frac{k(k-1)}{2} > (\alpha + 2n)(k-1) + n = d$, which implies that $RS(K, v_i) > d$. The above claim implies that, in any vote $v_i$, none of the candidates in $K$ appears in $X_i$, $Y_i$, or $Z_i$. Thus, all elements of $K$ are from $C$. Next, we show that, in each vote $v_i$, at least one of the candidates in $K$ appears in $\mathbb{C}[c_i]$. If this is not true, then there is a vote $v_i$ such that all the candidates in $K$ appear in $C \setminus \mathbb{C}[c_i]$. Then, $npos(K, v_i) \geq (n + \alpha)k > (\alpha + 2n)(k - 1) + n = d$, a contradiction. The above claim implies that the vertex corresponding to candidate $c_i$ must be adjacent to at least one of the vertices that correspond to the candidates in $K$. In other words, every vertex $v_i \in \mathcal{V}$ must be in at least one of the closed neighborhoods $\mathbb{N}[v'_1], \dots, \mathbb{N}[v'_k]$. By definition, set $\mathcal{V}'$ forms a dominating set of graph $\mathcal{G}$. $\square$

### 3.2.2 Parameter $d$: FPT

THEOREM 4. *RS-Minimax-Voting can be solved in $O^*\left(4^d\right)$*

*time and thus is fixed-parameter tractable with respect to the parameter $d$.*

PROOF. Let $v_i$ be an arbitrary vote in $V$. Note that all candidates $c \in C$ with $pos(c, v_i) > k + d$ cannot be in $K$, since otherwise, $npos(c, v_i)$ would be greater than $d$, which implies that $RS(K, v_i) > d$. In the following, we distinguish two cases:

$k \leq d$: In this case, we check for every size-$k$ subset $K$ of the first $k + d$ candidates in vote $v_i$ whether it satisfies $\max_{j=1,\ldots,n} RS(K, v_j) \leq d$. The time complexity is $O^*\left(\binom{d+k}{k}\right) = O^*\left(\binom{d+k}{d}\right) = O^*\left(\binom{2d}{d}\right) = O^*\left(4^d\right)$.

$k > d$: We claim that the first $k - d$ candidates in vote $v_i$ must be in $K$. For the purpose of contradiction, suppose that at least one of the first $k - d$ candidates in $v_i$ is not in $K$. Then at least $k - (k - d - 1) = d + 1$ candidates, whose positions are greater than $k - d$, must be in $K$. This would imply that $RS(K, v_i) > d$. Based on the above claim, all possible solutions have to contain the first $k - d$ candidates in $v_i$ and $k - (k - d) = d$ candidates from the candidates whose positions in vote $v_i$ are greater than $k - d$ and at most $k + d$. The brute-force approach leads to a time complexity of $O^*\left(\binom{2d}{d}\right) \leq O^*\left(4^d\right)$. □

### 3.2.3 Parameters $k$ and $n$: W[1]-hard

THEOREM 5. *RS-Minimax-Voting is W[1]-hard with respect to the parameters $k$ and $n$.*

PROOF. We reduce from the Multi-Colored Clique (MCC) problem, which, given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and an integer $k' \geq 0$, where $\mathcal{V}$ can be partitioned into $k'$ subsets and all edges in $\mathcal{E}$ are between different subsets, asks for a clique which contains exactly one vertex from each of the subsets. MCC is known to be W[1]-hard with respect to $k'$, even if all subsets have the same cardinality $n$ [14]. Moreover, it is not hard to prove the following claim:

**Claim.** MCC remains W[1]-hard with $k'$ as parameter even if

1. all subsets have the same cardinality $n$.
2. there exists a non-constant integer $D$, such that for each two subsets $S_1$ and $S_2$ and each vertex $v \in S_1$, it holds that $|\mathbb{N}(v) \cap S_2| = D$, where $\mathbb{N}(v)$ is the *neighborhood* of vertex $v$, i.e., $\mathbb{N}(v) = \{v_i | \{v, v_i\} \in \mathcal{E}\}$.

The basic idea of the reduction is to construct $(k'-1) \cdot D + 1$ candidates for every vertex, one for the vertex itself and one for each of its neighbors. Then, we construct for each of the $k'$ subsets $A \subset \mathcal{V}$ a vote, denoted by $v_A$, with the candidates that represent the vertices from this subset being in the first $n$ positions. The distance $d$ is set properly to make sure that at least one of these $n$ candidates needs to be in the committee, encoding that at least one vertex from this subset has to be in the clique. Moreover, we construct $3(k' - 1)$ votes for each subset $A$, three votes for each subset $B$ with $A \neq B$. In the first vote $v^1_{A,B}$ for $B$, the candidates which represent the $B$-neighbors of the vertices in $A$ are placed in the first $n \cdot D$ positions. The distance $d$ guarantees that at least one of these $n \cdot D$ candidates has to be in the committee. Further, the other two votes $v^2_{A,B}$ and $v^3_{A,B}$ guarantee that the candidate selected due to $v_A$ and the candidate selected due to $v^1_{A,B}$ correspond to the same vertex in $A$. Finally, we construct four votes to check the adjacency of the vertices correpsonding to the candidates which have been selected due to $v_A$ and $v_B$ respectively. In the following we present the details of the reduction.

We also use $a_i$ to denote the vertex candidate representing the vertex $a_i \in A$. Further the neighbor candidate representing the neighbor $b_j \in B$ of $a_i$ is denoted as $a_i^{b_j}$. Let $C'$ be the set of candidates constructed for the vertices in $\mathcal{V}$. Then, $|C'| = k' \cdot n \cdot (D(k' - 1) + 1)$. We set the size $k$ of the committee equal to $k'^2$. The first subset of votes contains $k$ votes, one for each vertex subset. For instance, the vote $v_A$ for $A$ has $a_1 \succ a_2 \succ, \ldots, \succ a_n$ in the first $n$ positions. Between these $n$ candidates and other candidates in $C'$ there are $x := (k'^2 - 1)(|C'| - n) + D$ many "dummy" candidates; they are ordered arbitrarily. Let $d$ denote the distance upper-bound, which will be set in the following. Each dummy candidate can be in the first $d$ positions of only one vote and thus, can never be in a committe with distance at most $d$ to the votes. The remaining candidates in $C'$ are placed directly after the dummy candidates, with an arbitrary order among them. By setting the distance upper-bound $d$ equal to $(k'^2 - 1)(|C'| + x) + n + D$, we observe that if none of the first $n$ candidates in $C'$ is added to the committee, then the distance between $v_A$ and the resulting committee is at least $k'^2(x + n)$ which is clearly greater than $d$. Thus, at least one candidates from $a_1, \ldots, a_n$ has to be in the committee.

Next, we present the three votes $v^1_{A,B}$, $v^2_{A,B}$, and $v^3_{A,B}$. In the first $n \cdot D$ positions of $v^1_{A,B}$, we have $a_1^{b_{j_1}} \succ \ldots \succ a_1^{b_{j_D}} \succ a_2^{b_{l_1}} \succ \ldots \succ a_n^{b_{s_D}}$, where the first $D$ candidates represent the neighbors of $a_1$ in $B$, the next $D$ candidates the neighbors of $a_2$, and so on. After them, there are $y = x - (nD - n - D)/(k'^2 - 1)$ many dummy candidates, and the order among these dummy candidates is arbitrary. Without loss of generality, we may assume here $nD - n - D$ can be divided by $k'^2 - 1$. The remaining candidates from $C'$ are placed after the dummy candidates with an arbitrary order. By the same argrument as for $v_A$, at least one candidate in the first $n \cdot D$ positions has to be in the committee. Since we have $k' - 1$ such $v^1$-votes for every vertex subset and these $k' - 1$ $v^1$-votes have distinct candidates in their first $n \cdot D$ positions, all feasible committees contain exactly one candidate in the first $n \cdot D$ positions of each $v^1$-vote. That is, for each of the $k'$ vertex subsets $A \subseteq \mathcal{V}$, a feasible committee contains exactly one vertex candidate $a_i$ which corresponds to a vertex in $A$ and exactly one neighbor candidate $a_j^{b_l}$ with $b_l \in B$ for every other subset $B \neq A$.

The votes $v^2_{A,B}$ and $v^3_{A,B}$ are constructed to force $i = j$. Both votes $v^2_{A,B}$ and $v^3_{A,B}$ put the candidates in $C' \setminus (\{a_1, a_2, \ldots, a_n\} \cup \{a_i^{b_j} | a_i \in A\})$ in the first $|C'| - nD - n$ positions. Then, the next $z = ((k'^2 - 2)|C'| + (k'^2 - 1)x)/n + D + 2$ positions of $v^2_{A,B}$ are occupied by dummy candidates, followed by $a_1$, and then, again $z$ many dummy candidates, followed by $a_2$, till $a_n$ is placed in the $(|C'| - nD + n \cdot z)$-th position. After $a_n$, we have again $z$ many dummy candidates followed by the $D$ neighbor candidates of $a_n$, which correspond to the $B$-neigbhors of $a_n$; the order among these neighbor candidates is arbitrary. After these neighbor candidates, we have again $z$ many dummy candidates and then the neighbor candidates of $a_{n-1}$. The last neighbor candidate of $a_1$ is in the $(|C'| + 2n \cdot z)$-th position. The vote $v^3_{A,B}$ has almost the same construction as $v^2_{A,B}$ with the only exception that the order of the vertex candidetes of $A$ is reversed, that is, $a_n$ appears as the first vertex candidate, then $a_{n-1}$, and so on; the appearance order of the neighbor candidates is also reversed, first the neighbor candidates of $a_1$, then the ones of $a_2$, and the last one being a neighbor candidate of $a_n$. With these two votes, we can guarantee that if a vertex candidate $a_i$ is added to committee due to the vote $v_A$, then the vote $v^1_{A,B}$ must add one neighbor candidate of $a_i$ to the committee. By doing so, we have a committee with a distance of less than $|C'| - (D+1)n + n \cdot z + D = d$ to both $v^2_{A,B}$ and $v^3_{A,B}$. If the chosen neighbor candidate is not of $a_i$, then at least one of $v^2_{A,B}$ and $v^3_{A,B}$ has a distance of more than $(n+1) \cdot z > d$ to the committee.

Finally, we have for each vertex subset $2(k'-1)$ votes to check the adajency between the vertex candidate of this subset and other chosen vertex candidates, two votes for each other subset. For instance, we have the following two votes for $A \subseteq \mathcal{V}$ to check whether the vertex candidate added to the committee due to $v_A$ is adjacent with the vertex candidate of $B$. These two votes are very similar to $v^2_{A,B}$ and $v^3_{A,B}$. Both have the candidates in $C' \setminus (\{a_1, a_2, \ldots, a_n\} \cup \{b^{a_i}_j | a_i \in A\})$ at the beginning. Then, both have the $a_i$'s blocks, each consisting of dummy candidates and one vertex candidate at the end, but one in the ascending order of the indices and the other in the descending order. The difference lies in the blocks of neighbor candidates. Here, for a vertex candidate $a_i$, the corresponding block contains instead of the neighbor candidates $a^{b_l}_i$'s but the neighbor candidates of $b_l$, that is, $b^{a_i}_l$'s. Again, the blocks appear in one vote in ascending order of the indices of $a_i$'s, and the orther in descending order. By the same argument, if $a_i$ is in the committee, then at least one of $b^{a_i}_l$'s has to be in the committee, which means the corresponding vertices are adjecent. Since this is true for every pair of subsets, a committee of size $k$ with distance at most $d$ to all votes can be directly transformed to a clique of size $k'$ in the original graph. Moreover, the number of votes and the size of the committee depend only on $k'$, which completes the theorem. $\square$

## 3.3 Dist-Minimax-Voting

Since Dist-Minimax-Voting is a generalized version of RS-Minimax-Voting, one can easily conclude that this problem is W[2]-hard with the committee size $k$ as parameter and W[1]-hard with both $k$ and the number $n$ of votes as parameters. It remains to prove that the parameterization of the distance upper-bound leads to an FPT-result.

THEOREM 6. *With the distance upper-bound $d$ as parameter, Dist-Minimax-Voting is fixed-parameter tractable.*

PROOF. We distinguish two cases $d < k$ and $d \geq k$. If $d \geq k$, then we consider an arbitrary vote $v$ and apply a depth-bounded search tree approach to enumerate all possible committees which have distance at most $d$ to $v$. These feasible committees can then be checked against all other votes with the same time bound. The depth of the search tree is bounded by $k+d$ and thus, its size by $2^{k+d} < 4^d$. During the branching process, we maintain three sets $K$, $U$, and $N$ and initialize $K$ and $N$ as empty sets and $U = C$. The set $K$ contains the candidates which should be added to the committee, the set $N$ the candidates not to add to the committee, while the candidates in $U$ are still undecided. The branching is terminated, if $U = \emptyset$, $d < 0$, or $k = 0$.

At a node of the search tree, we compute the set $S = \{a \in U | \nexists b \in U \text{ with } b \succ_v a\}$ and consider an arbitrary candidate $a \in S$. Note that $U \neq \emptyset$ implies $S \neq \emptyset$. We branch into two subcases: 1) $N := N \cup \{a\}$ and $d := d - k$; 2) If $k > 0$, then add $a$ to $K$ and set $k := k - 1$. Moreover, for each candidate $b \in N$ with $b \succ_v a$ decrease $d$ by one;. If at one leaf, we have $k \geq 0$ and $d \geq 0$, then output $K$ as solution.

The branching is clearly complete. In the first subcase, $a$ is not added to the committee. Then, by the definition of $S$, we have $a \succ_v b$ or $a \sim_v b$ for each candidate $b$ which will be added to $K$. Therefore, it is safe to decrease parameter $d$ by $k$. If we add $a$ to the committee, the parameter $k$ should be decreased by one. Further, since for the candidates $b \in N$ with $b \succ_v a$ we only decreased the parameter $d$ by one at the node dealing with $b$, we need to decrease $d$ by one. Thus, the search tree algorithm is correct. Since in one subcase we decrease $k$ by one and in the other $d$ is decreased by at least one, the total running time is then bounded by $O^*(2^{k+d}) = O^*(4^d)$. This completes the proof for $k < d$.

Concerning the case $k > d$, we again apply a search tree approach to enumerate all feasible committees for a fixed vote $v$. Note that in each step of the branching, the parameter $d$ is also decreased accordingly. The branching is terminated once $k < 0$ or $d < 0$. First we compute $S_0 := \{a \in C \mid \nexists b \text{ with } b \succ_v a\}$ and then for $i > 0$, $S_i := \{a \in (C \setminus \bigcup_{j<i} S_j) \mid \exists b \in S_{i-1} \text{ with } b \sim_v a\}$. Let $l$ be the maximal index with $S_l \neq \emptyset$ and set $S_{l+1} = C \setminus \bigcup_{i \leq l} S_i$. We need the following claim to prove the correctness of our algorithm.

**Claim.** For all subsets $S_i$ and $S_j$, if $j \geq i+2$, then we have $a \succ_v b$ for all $a \in S_i$ and $b \in S_j$.

*Proof of claim*: We prove this claim by contradiction. If $a \succ_v b$ is not true, then by the definition of $S_i$, we have $b \succ_v a$. Clearly, $a \notin S_0$. Then, there exists a candidate $a_1 \in S_{i-1}$ with $a_1 \sim_v a$. Since $b \notin S_i$, we have either $b \succ_v a_1$ or $a_1 \succ_v b$. Since $a_1 \succ_v b$ would imply $a_1 \succ_v a$ by transitivity of the linear order, we have $b \succ_v a_1$. This argument can then be applied to a candidate in $S_{i-2}$, which leads to the conclusion that there is a candidate $c$ in $S_0$ with $b \succ_v c$, contradicting to the definition of $S_0$. This completes the proof of the claim. $\square$

Clearly, for each feasible committee $K$, there exists a minimal index $i$ with $S_i \setminus K \neq \emptyset$. Moreover, we can observe that $i \geq 1$: Suppose there is a size-$k$ committee $K$ with $S_0 \setminus K \neq \emptyset$. Let $a$ be a candidate in $S_0 \setminus K$. Then, for each candidate $b \in K$ we have then $a \succ_v b$ or $a \sim_v b$, due to the definition of $S_0$. The distance between $K$ and $v$ is then at least $k > d$. Therefore, we iterate from $i = 1$ to $i = l+1$ and assume that $N := S_i \setminus K \neq \emptyset$ and $S_j \subseteq K$ for all $j < i$. The set $K$ is initialized as $\bigcup_{j<i} S_j$, while the set $N$, the set of the candidates not to be added to $K$, is initialized as empty.

If $|S_i| \leq d$, then we branch on every subset $X$ of $S_i$ and assume $X$ is not in $K$. This branch adds $S_i \setminus X$ to $K$, $X$ to $N$, and decreases $d$ by $2 \cdot |\{(a,b) \mid a \in X \text{ and } b \in K \text{ with } a \succ_v b\}| + |\{(a,b) \mid a \in X \text{ and } b \in K \text{ with } a \sim_v b\}|$. Since for every candidate $a \in S_i$ there is a candidate $b \in S_{i-1} \subseteq K$ with $a \sim_v b$, this branching leads to a $|S_i|$-bit branching vector with all entries being at least one.

If $|S_i| > d$, then we first consider the subset $S^0_i := \{a \in S_i \mid \nexists b \in S_i \text{ with } a \succ_v b\}$. If $|S^0_i| > d$, then we can conclude that there is no feasible committee $K$ with $S_i \setminus K \neq \emptyset$: None of the candidates in $S^0_i$ can be in $S_i \setminus K$, since for each candidate $a \in S_i$ there is a candidate $b \in S_{i-1} \subseteq K$ with $b \sim_v a$ and for every two candidates $a, b \in S^0_i$, we have $a \sim_v b$. If $S_i^0 \subseteq K$, then by $|S_i^0| > d$, no candidate of $S_i$ can be in $S_i \setminus K$ due to the definition of $S_i^0$. Next, we compute the set $S_i^1 := \{a \in (S_i \setminus S_i^0) | \nexists b \in S_i \setminus S_i^0 \text{ with } a \succ_v b\}$. By transitivity of the linear order and the facts that $|S_i^0| \leq d$ and $|S_i| > d$, $S_i^1 \neq \emptyset$. If $|S_i^0 \cup S_i^1| > d$, then $(S_i^0 \cup S_i^1) \setminus K \neq \emptyset$, since each candidate $a$ in $S_i \setminus K$ has $a \succ_v b$ or $a \sim_v b$ with each $b \in (S_i^1 \cup S_i^0)$ and thus, $(S_i^0 \cup S_i^1) \setminus K = \emptyset$ would result in that the distance between $K$ and $v$ exceeds $d$. With the same argument, we can conclude that for all $j$'s, if $|\bigcup_{l \leq j} S_i^l| > d$, then $(\bigcup_{l \leq j} S_i^l) \setminus K \neq \emptyset$. Let $j$ be the minimal index with $|\bigcup_{l \leq j} S_i^l| > d$. Since $|S_i| > d$, such an index $j$ must exist. Thus, $(\bigcup_{l \leq j-1} S_i^l) \setminus K \neq \emptyset$ or $S_i^j \setminus K \neq \emptyset$. By the same argument as for $S_i^0$, if $(\bigcup_{l \leq j-1} S_i^l) \setminus K = \emptyset$, then $|S_i^j| \leq d$. We apply here a branching into at most $2k$ subcases: If $|S_i^j| > d$, then branch into at most $2d$ subcases; each adds one candidate $a$ in $\bigcup_{l \leq j-1} S_i^l$ to $N$ and decreases $d$ by $2 \cdot |\{b \in K \mid a \succ_v b\}| + |\{b \in K \mid a \sim_v b\}|$. If $|S_i^j| \leq d$, then branch into at most $d$ subcases; each adds one candidate $a$ in $\bigcup_{l \leq j} S_i^l$ to $N$ and decreases $d$ accordingly. Again, since for every candidate $a \in S_i$ there is a candidate $b \in S_{i-1} \subseteq K$ with $a \sim_v b$, this branching leads to a branching vector better than the vector with $2d$ 1-entries.

After all candidates in $S_i$ have been added to either $K$ or $N$, we recompute the partition of the remaining candidates. While setting $S_0 := K$, we recompute $S_1 := \{a \in C \setminus (K \cup N) \mid \exists b \in S_0 \text{ with } b \sim_v a\}$. Observe that all candidates $a$ in $S_1$ have $b \succ_v a$ or $b \sim_v a$ with all candidates $b \in N$. Therefore, adding a candidate $a \in S_1$ to $K$ will decrease $d$ by at least one. Be the definition of $S_1$, adding $a$ to $N$ decreases $d$ by at least one too. This leads to a $(1,1)$-branching. We can then repeat this branching till $S_1 = \emptyset$. Note that every time we add a candidate to $N$ or $K$ we recompute $S_1$.

At this point, we have that all candidates $a$ in $C \setminus (K \cup N)$ satisfy $b \succ_v a$ or $b \sim_v a$ for all $b \in N \cup K$. If $|K| < k$ we have to add some of the remaining candidates to $K$. We now further partition these remaining candidates into subsets: $S_0 := \{a \in C \setminus (K \cup N) \mid \nexists b \in C \setminus (K \cup N) \text{ with } b \succ_v a\}$ and for $j > 0$, $S_j := \{a \in C' \setminus (N \cup \bigcup_{l<j} S_l) \mid \nexists b \in (N \cup \bigcup_{l<j} S_l) \text{ with } b \succ_v a\}$. Note that none of the vertices in $S_j$ with $j \geq d$ can be added to $K$ without violating the distance upper-bound $d$. Moreover, consider a $S_j$ with $j < d$. If $|S_j| > d$, then none of the candidates in $\bigcup_{l \geq j} S_l$ can be added to $K$. Therefore, we have at most $d^2$ possible candidates to add to $K$. Since all candidates $a$ in $C \setminus (K \cup N)$ satisfy $b \succ_v a$ or $b \sim_v a$ for all $b \in N$, we can add at most $d$ candidates to $K$. Thus, we have at most $\binom{d^2}{d}$ possibilities.

Altogether, we have at most $m \cdot \binom{d^2}{d}$ many possible committees which have distance at most $d$ to the vote $v$. All these committees can be enumerated in $O^*(\binom{d^2}{d})$ time. To verify whether a committee has distance at most $d$ to all other votes can be done in polynomial time, which completes the proof. $\square$

# 4. SPECIAL CASES OF HD-MINIMAX -VOTING

## 4.1 $V$ satisfies VI

We present a polynomial-time algorithm (Algorithm 1) to solve HD-Minimax-Voting for the special case that the input $V$ satisfies VI.

Our algorithm adopts a greedy strategy. Starting with an empty partial committee, it gradually expands the partial committee by processing the votes, one by one, following the VI order of the votes. In each step, it adds some candidates to the partial committee, without violating the distance upper-bound between the current vote and the partial committee. When doing this, the algorithm greedily selects such new candidates that are included in as many as possible votes that have not been processed so far. The algorithm stops, if either the distance between the current vote and the current partial committee exceeds the upper-bound, or the partial committee has $k$ candidates.

Theorem 7. *Given a HD-Minimax-Voting instance $(C, V, k, d)$, if $V$ satisfies VI, then Algorithm 1 can find a solution in $O(nm \log m)$ time.*

**Algorithm 1:** Algorithm for HD-Minimax-Voting in case that $V$ satisfies VI.

**Input** : List of votes $V = \{v_1, \ldots, v_n\}$ over candidate set $C = \{c_1, \ldots, c_m\}$, an order of the votes $v_1 \sqsubseteq \cdots \sqsubseteq v_n$ with respect to which $V$ satisfies VI, integers $d$ and $k$.
**Output**: A committee $K$ with $\max_{i=1,\ldots,n} HD(K, v_i) \leq d$ if it exists, and NOT FOUND otherwise.

```
K ← ∅;
for i ← 1 to n do
    X ← v_i \ K;
    if |K ⊕ v_i| + k − |K| > d then
        r ← ⌈(|K ⊕ v_i| + k − |K| − d) / 2⌉;
        if r > k − |K| or r > |X| then
            return NOT FOUND
        Sort the candidates c ∈ X in the non-increasing
          order of the number of votes that contain c
          and are after v_i in the order v_1 ⊑ ··· ⊑ v_n;
        Add the first r elements in the sorted X to K;
if k > |K| then
    Add arbitrary k − |K| candidates from C \ K to K;
return K;
```

## 4.2 $V$ satisfies CI

We present a polynomial-time algorithm (Algorithm 2) to solve HD-Minimax-Voting for the special case that the input $V$ satisfies CI.

Since $V$ satisfies CI, each vote occupies a consecutive block of the given CI order of the candidates. For a vote $v \in V$, we call the first and last candidates of the block its *head* and *tail*, respectively. Our algorithm first sorts the votes in $V$ according to the ascending order of the positions of their heads. If two votes have the same head position, then the vote with smaller weight lies in front of the other. Algorithm 2 adopts a similar greedy strategy as Algorithm 1. The main difference lies in how they select new candidates to modify the current partial committee.

Theorem 8. *Given a HD-Minimax-Voting instance*

*$(C, V, k, d)$, if $V$ satisfies CI, then Algorithm 2 can find a solution in time $O(nm \log m)$.*

**Algorithm 2:** Algorithm for $\delta$-Minimax-Voting in case that $V$ satisfies CI.

**Input** : List of votes $V = \{v_1, \ldots, v_n\}$ over candidate set $C = \{c_1, \ldots, c_m\}$, an order of the candidates $c_1 \lhd \cdots \lhd c_m$ with respect to which $V$ satisfies CI, integers $d$ and $k$.

**Output**: A committee $K$ with $\max_{i=1,\ldots,n} HD(K, v_i) \leq d$ if it exists, and NOT FOUND otherwise.

$V' \leftarrow V$, $V \leftarrow$ empty list, $V'' \leftarrow$ empty list;
**for** $i \leftarrow 1$ **to** $m$ **do**
  **if** *$V'$ is not empty* **then**
    $V'' \leftarrow$ all votes in $V'$ that contain $c_i$;
    $V' \leftarrow V' \setminus V''$;
  **if** *$V''$ is not empty* **then**
    Sort the votes in $V''$ in the non-decreasing order of their weights;
    Add $V''$ at the end of $V$;
$K \leftarrow \emptyset$;
**for** $i \leftarrow 1$ **to** $n$ **do**
  $X \leftarrow v_i \setminus K$;
  **if** $|K \oplus v_i| + k - |K| > d$ **then**
    $r \leftarrow \left\lceil \frac{|K \oplus v_i| + k - |K| - d}{2} \right\rceil$;
    **if** $r > k - |K|$ *or* $r > |X|$ **then**
      **return** NOT FOUND
    $Y \leftarrow$ all candidates in $K$ which lie after the tail of $v_i$ in the order $c_1 \lhd \cdots \lhd c_m$;
    Remove from $K$ $\min\{r, |Y|\}$ many candidates that have the minimal indices in the order $c_1 \lhd \cdots \lhd c_m$;
    Add into $K$ such $r$ elements in $X$ that have the maximal indices in the order $c_1 \lhd \cdots \lhd c_m$;
**if** $k > |K|$ **then**
  Add arbitrary $k - |K|$ candidates from $C \setminus K$ to $K$;
**return** $K$;

# 5. FUTURE RESEARCH DIRECTION

Amanatidis et al. [1] proposed a mixture of the minisum and minimax versions of approval voting, where an ordered weighted averaging operator balances the total distance of the committee to all votes and its maximal distance to individuals. They proved that for most relevant cases, the winner determination problem under the mixture model turns out to be NP-hard. It would be interesting to explore its parameterized complexity with respect to the parameters considered in this paper.

Aziz et al. [3] studied the complexity of several variants of the approval voting rule and among others, they proved that the winner determination problem for the proportional approval voting is fixed-parameter intractable with respect to the size of committee. The parameterized complexity of this problem with respect to other parameters and of other variants remains unexplored, a promising research direction.

## Acknowledgments

We are grateful to the anonymous reviewers of AAMAS'2016, whose comments greatly improve the readability of this paper.

## REFERENCES

[1] G. Amanatidis, N. Barrot, J. Lang, E. Markakis, and B. Ries. Multiple referenda and multiwinner elections using hamming distances: Complexity and manipulability. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*, pages 715–723, 2015.

[2] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. Justified representation in approval-based committee voting. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, 2015.

[3] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, and T. Walsh. Computational aspects of multi-winner approval voting. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*, pages 107–115, 2015.

[4] D. Baumeister. Personal communication, 2015.

[5] D. Baumeister and S. Dennisen. Voter dissatisfaction in committee elections. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*, pages 1707–1708, 2015.

[6] D. Baumeister, S. Dennisen, and L. Rey. Winner determination and manipulation in minisum and minimax committee elections. In *Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT'15)*, volume 9346 of *LNCS*, pages 469–485. Springer, 2015.

[7] N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier. Studies in computational aspects of voting - A parameterized complexity perspective. In *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 318–363. Springer, 2012.

[8] S. J. Brams, D. M. Kilgour, and M. R. Sanver. A minimax procedure for electing committees. *Public Choice*, 132:401–420, 2007.

[9] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[10] R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, and G. J. Woeginger. Parameterized algorithmics for computational social choice: Nine research challenges. *Tsinghua Science and Technology*, 19(4):358–373, 2014.

[11] I. Caragiannis, D. Kalaitzis, and E. Markakis. Approximation algorithms and mechanism design for minimax approval voting. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, 2010.

[12] V. Conitzer. Making decisions based on the preferences of multiple agents. *Commun. ACM*, 53(3):84–94, 2010.

[13] R. G. Downey and M. R. Fellows. *Parameterized complexity.* Springer, 1999.

[14] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity.* Springer, 2013.

[15] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. In *Proceedings of International conference on Autonomous Agents and Multi-Agent Systems (AAMAS'14)*, pages 53–60, 2014.

[16] E. Elkind and M. Lackner. Structure in dichotomous preferences. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*, pages 2019–2025, 2015.

[17] P. Faliszewski, E. Hemaspaandra, L. A. 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.

[18] D. Felsenthal. On combining approval with disapproval voting. *Behavioral Science*, 34:53–60, 1989.

[19] J. Flum and M. Grohe. *Parameterized complexity theory.* Springer, 2006.

[20] R. LeGrand, E. Markakis, and A. Mehta. Some results on approximating the minimax solution in approval voting. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'07)*, 2007.

[21] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of operations research*, 8(4):538–548, 1983.

[22] R. Meir, A. D. Procaccia, and J. S. Rosenschein. A broader picture of the complexity of strategic behavior in multi-winner elections. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*, pages 991–998, 2008.

[23] R. Meir, A. D. Procaccia, J. S. Rosenschein, and A. Zohar. Complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, 2008.

[24] N. Misra, A. Nabeel, and H. Singh. On the parameterized complexity of minimax approval voting. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*, pages 97–105, 2015.

[25] R. Niedermeier. *Invitation to fixed-parameter algorithms.* Oxford University Press, 2006.

[26] S. Obraztsova, Y. Zick, and E. Elkind. On manipulation in multi-winner elections based on scoring rules. In *Proceedings of International conference on Autonomous Agents and Multi-Agent Systems (AAMAS'13)*, pages 359–366, 2013.