



Parameterized complexity of multiwinner determination: more effort towards fixed-parameter tractability

Yongjie Yang¹ · Jianxin Wang^{2,3}

Accepted: 26 May 2023
© The Author(s) 2023

Abstract

We study the parameterized complexity of winner determination problems for three prevalent k -committee selection rules, namely the minimax approval voting (MAV), the proportional approval voting (PAV), and the Chamberlin–Courant’s approval voting (CCAV). It is known that these problems are computationally hard. Although they have been studied from the parameterized complexity point of view with respect to several natural parameters, many of them turned out to be $W[1]$ -hard or $W[2]$ -hard. Aiming at obtaining plentiful fixed-parameter algorithms, we revisit these problems by considering more natural single parameters, combined parameters, and structural parameters.

Keywords Multiwinner voting · Fixed-parameter tractability · Minimax approval voting · Proportional approval voting · Chamberlin–Courant’s approval voting · Treewidth · $W[1]$ -hard

1 Introduction

Committee selection rules (a.k.a. multiwinner voting rules) have received a considerable amount of attention recently due to their broad applications in social choice, multi-agent systems, recommendation systems, etc. [31, 39, 40, 49]. How efficiently winning candidates with respect to a committee selection rule can be calculated is one of the most significant criteria to evaluate the applicability of these rules. Many rules, such as STV, Bloc, k -Borda, approval voting, satisfaction approval voting, admit polynomial-time algorithms for computing winners [3, 28]. However, there are also salient committee selection rules with respect to

A 3-page extended abstract of the paper appeared in the proceedings of the *17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)* [80].

✉ Yongjie Yang
yyongjiecs@gmail.com

Jianxin Wang
jxwang@mail.csu.edu.cn

¹ Chair of Economic Theory, Saarland University, 66123 Saarbrücken, Germany

² School of Computer Science and Engineering, Central South University, Changsha 410083, China

³ Xiangjiang Laboratory, Changsha 410205, China

which winners are NP-hard to compute. Among them are particularly the minimax approval voting (MAV) [11, 50], the proportional approval voting (PAV) [3, 43, 70], and the Chamberlin–Courant’s approval voting (CCAV) [18, 61]. Nevertheless, as these rules possess their own merits in many other aspects [2, 33, 43, 44, 49], researchers investigated the parameterized complexity of winner determination problems under these rules, with the hope of procuring as many fixed-parameter tractability results as possible. While many realistic parameters have been considered in the literature so far, there are still many relevant but underappreciated parameters. This paper aims to take a further step towards breaking the complexity barrier against the applicability of the above-mentioned three approval-based rules by extensively expanding the set of meaningful parameters leading to FPT-algorithms for the winner determination problems. Notably, in addition to many traditional parameters, we also study several structural parameters of the incidence graphs of approval-based elections. The incidence graph of an approval-based election is a bipartite graph whose vertex set is the candidate set union the voter set, and there is an edge between a candidate and a voter if and only if this voter approves the candidate. To date, less is known about whether some structural parameters of the incidence graphs such as the treewidth and the size of a maximum matching lead to some FPT-algorithms. These parameters are intimately connected to several important single parameters. For example, they are lower bounds of both the number of candidates m and the number of voters n , and hence any FPT-algorithm with respect to these two structural parameters directly carry over to m and n .

Organization. In Sect. 2, we give definitions and notations used in the paper. Then, we elaborate on related works and summarize our main results in Sect. 3. Our concrete results are encapsulated in Sects. 4–6. Specifically, Sect. 4 studies single parameters, Sect. 5 explores combinations of the single parameters occurring in Sect. 4 and Sect. 6 focuses on the aforementioned structural parameters. For an overview of our main results, we refer to Table 1. We complete the paper by recapping our contributions and laying out several promising avenues for future research in Sect. 7.

2 Preliminaries

In this section, we give essential notions related to our study.

2.1 Elections

We study *approval-based multiwinner voting*. In this setting, an *election* is a tuple $E = (C, V)$ where C is a set of candidates and V is a multiset of votes. Each vote of V is cast by a voter and is defined as a subset of C . In this paper, we interchangeably use the terms vote and voter. We say that a vote v *approves* a candidate c if $c \in v$. For each candidate c , $V(c)$ denotes the set of votes in V approving c . Let k be a nonnegative integer. A *k-set* is a set of cardinality k . A *committee* (respectively, *k-committee*) is a subset (respectively, *k-subset*) of candidates. A *k-committee selection rule* maps each election (C, V) and every nonnegative integer k such that $k \leq |C|$ to a collection of *k-committees* of C , *winning k-committees* of (C, V) under this rule.

The *Hamming distance* between two sets v and v' is defined as

$$\mathcal{H}(v, v') = |v \setminus v'| + |v' \setminus v|.$$

We study the following *k-committee selection rules*.

Table 1 A summary of the parameterized complexity of τ -MULTIWINNER

		single parameters				
	d	m	n	k	\bar{k}	(Δ_V, Δ_C)
MAV	<u>FPT</u> [57]	<u>FPT</u> [57]	<u>FPT</u> [57]	<u>W[2]-h</u> [57]	<u>W[1]-h*</u> (Cor. 1)	$(\geq 2, \geq 3)$: NP-h [50] others: P (Cor. 5, Thm. 4)
CCAV	$2^{O(d)}$ (Cor. 2)	<u>FPT</u> [4]	n^d [4] $2^{O(\omega)}$ (Cor. 3)	<u>W[2]-h</u> [4] <u>W[1]-h*</u> (Thm. 7)	<u>W[1]-h*</u> (Thm. 2)	$(\geq 2, \geq 3)$: NP-h [62] others: P (Cor. 5, Thm. 4)
PAV	open	FPT	FPT (Thm. 1)	<u>W[1]-h*</u> [3]	<u>W[1]-h*</u> (Thm. 3)	$(\geq 2, \geq 3)$: NP-h [3] $(\geq 3, 2)$: open others: P (Cor. 5, Thm. 5)
	combined parameters					structural parameters
MAV	$k + \Delta_C$ FPT (Thm. 6)	$k + \Delta_V$ FPT (Cor. 7)	$\bar{k} + \Delta_C$ FPT (Thm. 8)	$\bar{k} + \Delta_V$ <u>W[1]-h*</u>	$d + \Delta_V$ FPT	ω $\omega + k$ FPT (Thm. 12)
CCAV	FPT (Cor. 6)	<u>W[1]-h*</u>	FPT (Thm. 9)	<u>W[1]-h*</u>	FPT	4^{ω} (Thm. 11)
PAV	$k + \Delta_C + \Delta_V$ FPT (Cor. 8)	<u>W[1]-h*</u> [3]	open	<u>W[1]-h*</u>	FPT (Thm. 10)	$\omega + k$ FPT (Thm. 12)

Our results are in boldface. Additionally, ω is the treewidth and α is the size of a maximum matching of the incidence graph of a given election. Hardness results marked by the \star symbol mean that they hold even if every vote approves two candidates. Results without any marks mean that they are either trivial or implied by other results in the table. Underlined FPT-results mean that the corresponding problems do not admit any polynomial kernels assuming certain standard complexity hypothesis

- MAV** In general, MAV selects k -committees as close as possible to every vote, where the closeness is measured by the Hamming distance. Precisely, the MAV score of a committee w with respect to (C, V) is $\text{MAV}(V, w) = \max_{v \in V} \mathcal{H}(v, w)$. MAV selects k -committees with the minimum MAV score.
- CCAV** A vote v is satisfied with a committee w if and only if at least one of v 's approved candidates is contained in w , i.e., $v \cap w \neq \emptyset$. Candidates in $v \cap w$ are regarded as representatives of v in w . The CCAV score of a committee w with respect to (C, V) is $\text{CCAV}(V, w) = |\{v \in V : v \cap w \neq \emptyset\}|$. CCAV selects k -committees with the maximum CCAV score.¹
- PAV** The PAV score of a committee w with respect to an election (C, V) is $\text{PAV}(V, w) = \sum_{v \in V, v \cap w \neq \emptyset} \sum_{i=1}^{|v \cap w|} \frac{1}{i}$. PAV selects k -committees with the maximum PAV score.

Note that by the above definitions, $\text{CCAV}(V, \emptyset) = \text{PAV}(V, \emptyset) = 0$ holds.

For each $\tau \in \{\text{MAV}, \text{CCAV}, \text{PAV}\}$, we study the following problem.

τ -MULTIWINNER

Input: An election $E = (C, V)$, an integer $k \leq |C|$, and a rational number d .

Question: Is there a k -committee $w \subseteq C$ such that $\tau(V, w) \leq d$ for $\tau = \text{MAV}$, and $\tau(V, w) \geq d$ for $\tau \in \{\text{CCAV}, \text{PAV}\}$?

Throughout this paper, we study the following single parameters and consistently use the corresponding notations given below.

- $m = |C|$.
- $n = |V|$.
- d : the threshold score of a desired committee.
- k : size of winning committees.
- $\bar{k} = m - k$.
- $\Delta_V = \max_{v \in V} |v|$ is the maximum number of candidates a vote approves.
- $\Delta_C = \max_{c \in C} |V(c)|$ is the maximum number of votes approving a candidate in common.

The first four parameters are natural and have been substantially covered in the literature. (See Sect. 3 for the details.) The study of \bar{k} is motivated by the observation that in many real-life decision-making scenarios winners are picked by eliminating a small number of losers, or more relevantly, many decision-making processes are directly designed to select losers other than selecting winners. The last two parameters Δ_V and Δ_C have been either explicitly or implicitly studied in the literature. In many real-world applications voters are allowed to approve only a few candidates. In some other scenarios, voters are cognitively limited or time constrained so that they are only able to evaluate a small number of candidates. In these cases, Δ_V is relatively small. Regarding the parameter Δ_C , observe that it is no greater than n . So, it is small whenever n is small. Moreover, as our goal is to provide a landscape of the parameterized complexity of τ -MULTIWINNER as complete as possible,

¹ CCAV is a special rule of the class of Chamberlin–Courant's rules. See [28] for further discussions.

it is theoretically significant to study this parameter so as that we can offer results for combined parameters including Δ_C .

2.2 Graphs

We assume the reader is familiar with the basic concepts of graph theory, and refer to [12, 73] for notions in graph theory used but not defined in the paper. We only reiterate some basic notions below. A *graph* is a pair (N, A) where N is a set of vertices and A is a set of edges over N . Each *edge* is defined as an unordered pair of vertices, and between every two vertices there can be at most one edge. A *multigraph* is a generalization of a graph where between every two vertices there may exist multiple edges, and there may exist loops on vertices. A *hypergraph* is a generalization of a graph where every edge is a subset of vertices. A *multihypergraph* is a generalization of a hypergraph so that there may exist multiple edges consisting of the same vertices.

A *matching* of a graph $G = (N, A)$ is a subset M of A such that no two edges in M share a common vertex. A vertex v is *saturated* by M if v is a vertex in some edge in M . A *maximum matching* of G is a matching with the maximum cardinality among all matching of G . We use $\alpha(G)$ to denote the size of a maximum matching of G .

2.3 Parameterized complexity

A parameterized problem is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. A parameterized problem can be either *fixed-parameter tractable* (FPT) or *fixed-parameter intractable*. In particular, a parameterized problem is FPT if there is an algorithm which correctly determines for each instance (I, κ) of the problem whether (I, κ) is a Yes-instance in time $O(f(\kappa) \cdot |I|^{O(1)})$, where f is a computable function and $|I|$ is the size of I . Fixed-parameter intractable problems are further classified into many classes including W[1]-hard, W[2]-hard, etc. For greater details on parameterized complexity theory, we refer to [22, 25, 27].

3 Related works and our contributions

In this section, we discuss some important related works and outline our main contributions.

3.1 Single parameters

τ -MULTIWINNER has many natural parameters inherent in its definition, say, the seven single parameters listed at the end of Sect. 2.1. All these seven parameters except \bar{k} have been explicitly or implicitly investigated in the literature prior to our work.

First, it is easy to see that MAV-MULTIWINNER, CCAV-MULTIWINNER, and PAV-MULTIWINNER are FPT with respect to the parameter m . Misra, Nabeel, and Singh [57] proved that MAV-MULTIWINNER is FPT with respect to the parameters d and n , but becomes W[2]-hard when parameterized by k . Betzler, Slinko, and Uhlmann [4] proved that CCAV-MULTIWINNER is FPT with respect to the parameter n , but turned out to be W[2]-hard when k is the parameter. Moreover, they considered a dual parameter $R = n - d$. They proved that CCAV-MULTIWINNER is NP-hard even when $R = 0$, but presented an FPT-algorithm with

respect to the combined parameter $k + R$.² Aziz et al. [3] proved that PAV-MULTIWINNER is $W[1]$ -hard with respect to k even if every voter approves two candidates.

We complement these results as follows. First, we close some gaps and improve an FPT-algorithm. Concretely, we propose an FPT-algorithm for PAV-MULTIWINNER when parameterized by n . We also observe that CCAV-MULTIWINNER is equivalent to the PARTIAL HITTING SET problem, and as a consequence of a previous result for the PARTIAL HITTING SET problem, CCAV-MULTIWINNER can be solved in $O^*(2^{O(d)})$ time or in $O^*(2^{O(n)})$ time.³ It should be noted that our new observation-based algorithm substantially improves the previous best FPT-algorithm for CCAV-MULTIWINNER parameterized by n studied in [4] which runs in time $O^*(n^n)$. Second, we study the parameter $\bar{k} = m - k$, the number of nonwinning candidates. We show that MAV-MULTIWINNER, CCAV-MULTIWINNER, and PAV-MULTIWINNER are all $W[1]$ -hard with respect to this parameter, even when every voter approves two candidates. Third, from previous results by other researchers, we achieve numerous dichotomy results with respect to the two natural parameters Δ_V and Δ_C . It is known that MAV-MULTIWINNER, CCAV-MULTIWINNER, and PAV-MULTIWINNER are already NP-hard when $\Delta_V = 2$ and $\Delta_C = 3$ [3, 50, 62]. We prove that MAV-MULTIWINNER and CCAV-MULTIWINNER become polynomial-time solvable if $\Delta_V \leq 1$ or $\Delta_C \leq 2$, and PAV-MULTIWINNER becomes polynomial-time solvable if $\min\{\Delta_V, \Delta_C\} = 1$ or $\Delta_V = \Delta_C = 2$.

3.2 Combined parameters

Obviously, if a problem is FPT with respect to a parameter κ , it is FPT with respect to any combined parameter including κ . As except PAV-MULTIWINNER with respect to d whose fixed-parameter tractability is open, MAV-MULTIWINNER, CCAV-MULTIWINNER, and PAV-MULTIWINNER are FPT with respect to m , n , and d , it only makes sense to study combinations of other parameters. As $k + \bar{k} = m$, τ -MULTIWINNER for $\tau \in \{\text{MAV}, \text{CCAV}, \text{PAV}\}$ is FPT with respect to $k + \bar{k}$. For MAV and CCAV, the remaining combinations of two single parameters are $k + \Delta_V$, $k + \Delta_C$, $\bar{k} + \Delta_V$, and $\bar{k} + \Delta_C$. We establish many FPT-results with respect to these combined parameters. Concretely, we obtain FPT-results for MAV-MULTIWINNER and CCAV-MULTIWINNER when parameterized by $k + \Delta_C$ and $\bar{k} + \Delta_C$. However, as MAV-MULTIWINNER, CCAV-MULTIWINNER, and PAV-MULTIWINNER are $W[1]$ -hard with respect to \bar{k} (even when every vote approves two candidates), they are $W[1]$ -hard when parameterized by $\bar{k} + \Delta_V$. For the parameter $k + \Delta_V$, we develop an FPT-algorithm for MAV-MULTIWINNER, but we show that CCAV-MULTIWINNER is $W[1]$ -hard even when every vote approves two candidates. Concerning PAV, a reduction by Aziz et al. [3] implies that PAV-MULTIWINNER is $W[1]$ -hard with respect to $k + \Delta_V$. We are unable to prove the FPT-membership of PAV-MULTIWINNER with respect to d , but we show that combining d and Δ_V leads to an FPT-result.

We would like to point out that Misra, Nabeel, and Singh [57] studied kernelization of MAV-MULTIWINNER with respect to the combined parameters $d + m$ and $n + k$, and showed

² A parameterized problem is FPT with respect to the combination of two parameters κ and κ' if it is solvable in time $O^*(f(\kappa, \kappa'))$ where f is a computable function in κ and κ' , or equivalently it is FPT with respect to $\kappa + \kappa'$.

³ $O^*(\cdot)$ is $O(\cdot)$ with polynomial factors being omitted.

that these problems do not admit any polynomial kernels unless $\text{CoNP} \subseteq \text{NP}/\text{Poly}$ ⁴. In addition, Liu and Guo [53] studied the combined parameter $k + n$ for some generalizations of MAV-MULTIWINNER. Betzler, Slinko, and Uhlmann [4] proved that CCAV-MULTIWINNER is $\text{W}[2]$ -hard with respect to the combined parameter of $R = n - d$ and k . Though not explicitly stated, their reduction actually implies that CCAV-MULTIWINNER does not admit any polynomial kernel with respect to both n and m , unless the polynomial hierarchy collapses to the third level.⁵

3.3 Structural parameters

Heretofore, the most widely studied structural parameters in the setting of multiwinner voting are based on various concepts of restricted preference domains such as single-peaked preferences and single-crossing preferences (see, e.g., [20]). Particularly, these parameters measure how far an election is away from a specific domain of preferences. In this paper, we study two structural parameters of incidence graphs of elections. Recall that the *incidence graph* of an election $E = (C, V)$ is a bipartite graph G_E with vertex set $C \cup V$ so that there is an edge between a candidate $c \in C$ and a vote $v \in V$ if and only if $c \in v$. We prove that CCAV-MULTIWINNER is FPT with respect to treewidth of incidence graphs, and MAV-MULTIWINNER and PAV-MULTIWINNER are FPT if we combine the treewidth and the parameter k . When parameterized by the size of maximum matchings of incidence graphs, we have FPT-algorithms for all three rules.

3.4 Other related works

In addition to the extensive effort made from the angle of parameterized complexity, much exploration on the complexity of τ -MULTIWINNER restricted to preference domains has been pursued over the past few years. Betzler, Slinko, and Uhlmann [4] developed polynomial-time algorithms for CCAV-MULTIWINNER in the single-peaked domain. This algorithm was subsequently extended to an FPT-algorithm with respect to the parameter single-peaked width by Cornaz, Galand, and Spanjaard [20]. Yu, Chan, and Elkind [81] studied the domain of single-peaked on trees, and obtained both polynomial-time algorithms and NP-hardness results for many variants of CCAV. One of their results [81, Theorem 4.1] is in essence an FPT-algorithm for CCAV-MULTIWINNER with respect to the combined parameter $m + k + \lambda$, where λ is the number of leaves of the underlying tree. A follow-up paper by Peters and Elkind [59] addressed several open questions left in [81]. Later, Elkind and Lackner [29] proposed 13 different restricted domains of dichotomous preferences, and obtained several polynomial-time algorithms and FPT-algorithms for τ -MULTIWINNER for τ being CCAV, MAV, and PAV with respect to the parameter Δ_V , and the combined parameter $k + d$, when the given elections fall into certain specific categories of their proposed domains. Liu and Guo [53] presented polynomial-time algorithms for MAV-MULTIWINNER in two of the domains proposed by Elkind and Lackner. Peters and Lackner [60]

⁴ A kernelization with respect to a parameter is a polynomial-time algorithm which transforms an instance into an equivalent instance with the size of the main part being bounded from above by a computable function of the parameter. See [22, Chapter 2] for more details.

⁵ These kernelization lower bounds follow from a reduction by Betzler, Slinko, and Uhlmann [4] and several techniques for establishing kernelization lower bounds delineated in [22, Chapter 14] and [24].

proposed the domain of single-peaked on a circle, an appealing generalization of the single-peaked domain. They provided a polynomial-time algorithm for CCAV-MULTIWINNER in this domain. Peters [58] observed a relationship between voting problems restricted to the single-peaked domain and totally unimodular integer linear programming. In view of this observation and the fact that totally unimodular integer linear programming is polynomial-time solvable [64], a variety of polynomial-time solvability results were obtained, including one for PAV-MULTIWINNER.⁶ Skowron et al. [68] complemented these results by showing that CCAV-MULTIWINNER is polynomial-time solvable restricted to the single-crossing domain. Clearwater, Puppe, and Slinko [19] then extended this result to the domain of single-crossing on trees. Yang [78] expanded several domains of Elkind and Lackner [29] to directed tree-embedded domains, and derived a number of polynomial-time algorithms for MAV-MULTIWINNER, CCAV-MULTIWINNER, and PAV-MULTIWINNER restricted to these domains.

As traditional approaches tackling NP-hard problems, approximation and heuristic algorithms for τ -MULTIWINNER where $\tau \in \{\text{MAV}, \text{CCAV}, \text{PAV}\}$ have also been perpetually reported over the past few years [16, 23, 35, 65, 66].

We would like to mention that the (parameterized) complexity of τ -MULTIWINNER for numerous ranking-based multiwinner voting rules τ has been considerably studied in the literature, too [4, 32, 42]. On top of that, the (parameterized) complexity of winner determination problems for several variants of multiwinner voting rules has been explored recently [13, 75].

Finally, we remark that, besides winner determination problems, investigating the complexity of many strategic voting problems, such as manipulation, control, and bribery, has gained increasing interest in recent years as well [14, 34, 76, 77]. We refer to [48] for a comprehensive survey of approval-based multiwinner voting where many other important issues not discussed above have been greatly elaborated on.

4 Single parameters

In this section, we investigate some predominant single parameters.

4.1 Parameters d , n , and \bar{k}

We start with an FPT-algorithm for PAV-MULTIWINNER with respect to n . Before presenting our algorithm, let us recall the main idea of an FPT-algorithm for MAV-MULTIWINNER with respect to n [57]. First, the candidates are partitioned into at most 2^n subsets, each consisting of all candidates approved by exactly the same votes. Then, what matters for solving the problem is only how many candidates from each subset are contained in a desired k -committee. Based on this observation, MAV-MULTIWINNER can be reduced to integer linear programming (ILP) by assigning to each subset defined above an integer variable. As the number of variables is bounded by 2^n , Lenstra's theorem [51] provides an FPT-algorithm for MAV-MULTIWINNER with respect to n . Although the FPT-algorithm with respect to n for CCAV-MULTIWINNER derived by Betzler, Slinko, and Uhlmann [4] is not ILP-based, it is easy to see that CCAV-MULTIWINNER also admits a similar ILP formulation.

⁶ We note that [60] integrates a conference paper with the same title appeared in AAAI 2017 and [58].

Unfortunately, this framework does not apply to PAV-MULTIWINNER. The reason is that the objective in this case is a nonlinear function. To overcome this obstacle, we resort to an FPT-framework proposed by Brederick et al. [15] for the MIXED INTEGER PROGRAMMING WITH SIMPLE PIECEWISE LINEAR TRANSFORMATIONS problem. In fact, we need only a special case of the problem defined below. For a vector $x \in \mathbb{Z}^p$ and an integer $i \in [p]$, we use x_i to denote the i -th component of x .

A real-valued function f is *concave* if for every x and y such that $x < y$ and every λ such that $0 \leq \lambda \leq 1$ it holds that

$$f(\lambda \cdot x + (1 - \lambda) \cdot y) \geq \lambda \cdot f(x) + (1 - \lambda) \cdot f(y).$$

Intuitively, a function is concave if for every x, y , and z such that $x \leq y \leq z$, the point $(y, f(y))$ is not below the straight line determined by the two points $(x, f(x))$ and $(z, f(z))$. A *piecewise linear concave function* is a piecewise linear function that is concave.

Integer Programming With Simple Piecewise Linear Transformations (IPWSPLT)

Input: A collection $\{f_{i,j} : i \in [p], j \in [q]\}$ of $p \cdot q$ piecewise linear concave functions, and a vector $b \in \mathbb{Z}^p$.

Question: Is there a vector $x \in \mathbb{Z}^q$ such that for every $i \in [p]$ it holds that

$$\sum_{j=1}^q f_{i,j}(x_j) \leq b_i? \quad (1)$$

The original problem MIPWSPLT studied by Brederick et al. [15] is more general in that it allows the existence of additional variables which may take nonintegral values, and allows the occurrences of both piecewise linear concave functions and piecewise linear convex functions simultaneously.

Lemma 1 ([15]) *IPWSPLT can be solved in time $O(\text{poly}(|I|, t) \cdot q^{2.5q+o(q)})$, where $|I|$ is the number of bits encoding the input, and t is the maximum number of pieces per function.*

We note that Lemma 1 still holds if in the definition of IPWSPLT the less than sign is replaced with the greater than sign or the equal sign in (1) for several $i \in [p]$ [15].

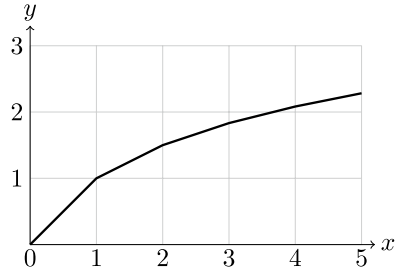
Now we are ready to present our FPT-algorithm for PAV-MULTIWINNER with respect to n . We provide indeed an algorithm for a more general problem called ANNOTATED PAV-MULTIWINNER. In this problem, we are given an election (C, V) , a subset $C' \subseteq C$ of candidates, an integer k such that $|C'| \leq k \leq |C|$, and a number d , and the question is whether there is a k -committee $w \subseteq C$ such that $C' \subseteq w \subseteq C$ and $\text{PAV}(V, w) \geq d$. Clearly, PAV-MULTIWINNER is a special case of ANNOTATED PAV-MULTIWINNER where $C' = \emptyset$. This generalization is afterward exploited in an algorithm presented in Sect. 6.

Theorem 1 *ANNOTATED PAV-MULTIWINNER is FPT with respect to n .*

Proof We prove Theorem 1 by reducing ANNOTATED PAV-MULTIWINNER to IPWSPLT.

Let $I = (E, C', k, d)$ be an instance of ANNOTATED PAV-MULTIWINNER, where $E = (C, V)$ and $C' \subseteq C$. Let $n = |V|$ be the number of votes. In what follows, we construct an integer programming formulation of ANNOTATED PAV-MULTIWINNER, prove the correctness of the reduction, and prove that the formulation is an instance of IPWSPLT. We create two types of variables. First, we create a variable x_v for each vote $v \in V$, indicating the number of v 's approved candidates in a desired k -committee. Second, for each $U \subseteq V$, let C_U

Fig. 1 An illustration of the piecewise linear concave function f in the proof of Theorem 1



be the set of candidates approved by all votes in U but disapproved by votes from $V \setminus U$, i.e., $C_U = \{c \in C : V(c) = U\}$. We create a variable x_U for each $U \subseteq V$ which indicates the number of candidates from C_U that are supposed to be in a desired k -committee. The constraints are as follows.

- (1) for each variable $x_U, U \subseteq V$, it holds that $|C' \cap C_U| \leq x_U \leq |C_U|$.
- (2) For each $v \in V$, we have $x_v = \sum_{U \in \mathcal{U} \subseteq V} x_U$.
- (3) As we aim to select a committee of cardinality k , we have $\sum_{U \subseteq V} x_U = k$.
- (4) For the last constraint, we need to define a piecewise linear concave function $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ as follows. First, $f(0) = 0$. Second, for each positive integer x , we define $f(x) = \sum_{i=1}^x \frac{1}{i}$. Third, for each real x such that $y < x < y + 1$ and y is a nonnegative integer, we define

$$f(x) = f(y) + (x - y) \cdot (f(y + 1) - f(y)).$$

Fig. 1 illustrates the function f . The last constraint is then $\sum_{v \in V} f(x_v) \geq d$, which ensures that the desired k -committee has PAV score at least d .

Now we show the correctness of the reduction, i.e., we show that the given instance I is a Yes-instance if and only if the above integer programming has a feasible solution.

(\Rightarrow) Assume that I is a Yes-instance, i.e., there exists a k -committee w so that $C' \subseteq w \subseteq C$ and $\text{PAV}(V, w) \geq d$. We assign to the variables constructed above the corresponding values, i.e., we let $x_U = |w \cap C_U|$ for each $U \subseteq V$, and let $x_v = |v \cap w|$ for each $v \in V$. Obviously, $|w \cap C_U| \leq |C_U|$. As $C' \subseteq w$, it holds that $|C' \cap C_U| \leq |w \cap C_U|$. Therefore, constraints defined in (1) hold. As for every two distinct $U, U' \subseteq V$, C_U and $C_{U'}$ are disjoint and, moreover, $\bigcup_{U \subseteq V} C_U = C$, it holds that

$$|v \cap w| = \sum_{U \subseteq V} |v \cap w \cap C_U| = \sum_{U \subseteq V} |w \cap C_U| = \sum_{U \subseteq V} x_U \tag{1}$$

for all $v \in V$, and $\sum_{U \subseteq V} |w \cap C_U| = |w| = k$. In other words, all constraints described in (2)–(3) are satisfied. Notice that by the definition of C_U , if $v \notin U$ then $C_U \cap v = \emptyset$, and if $v \in U$ then $C_U \subseteq v$. This ensures the correctness of Equality (1). Finally, the constraint given in (4) holds because $\sum_{v \in V} f(x_v)$ is exactly the PAV score of w with respect to V which is equal to or greater than d .

(\Leftarrow) Assume that the above integer programming has a feasible solution. We show below that I is a Yes-instance by constructing a desired k -committee w . Initially, let $w = \emptyset$. For every $x_U, U \subseteq V$, in the feasible solution, we arbitrarily select x_U candidates from C_U so that all candidates in $C' \cap C_U$ are selected, and add them into w . By the constraints in (1), these x_U candidates exist for each $U \subseteq V$. By the constraint in (3), we know that w consists

of k candidates. By the constraints in (2), x_v is exactly the number of candidates from w approved in v . As a result, $\sum_{v \in V} f(x_v)$ is exactly the PAV score of w with respect to V . Then, from the constraint in (4), it follows that $\text{PAV}(V, w) \geq d$. Finally, as $\bigcup_{U \subseteq V} C_U = C$, by the definition of w we know that C' is contained in w . Now we can conclude that I is a Yes-instance of ANNOTATED PAV-MULTIWINNER.

Next, we show that the above integer programming is an instance of IPWSPLT. To this end, we reiterate that IPWSPLT contains ILP as a special case. Constraints described in (1)–(3) are standard constraints (or can be transformed into standard form trivially) of ILP. It is easy to see that the function f defined in (4) is a piecewise linear concave function. Therefore, the above integer programming is an instance of IPWSPLT.

It remains to analyze the running time of the algorithm. The above reduction clearly takes FPT-time in n , and the number of variables is $n + 2^n$. Then, by Lemma 1 and the correctness of the reduction, ANNOTATED PAV-MULTIWINNER is FPT with respect to n . \square

We note that a similar FPT-algorithm for winners computation for a large class of ranking-based multiwinner voting rules has been also derived by Faliszewski et al. [32, Theorem 16].

Now we study the parameter $\bar{k} = m - k$, i.e., the number of candidates not in a desired k -committee. In regard to MAV, the NP-hardness proof by LeGrand [50] actually already implied that MAV-MULTIWINNER is W[1]-hard with respect to \bar{k} . Precisely, LeGrand’s NP-hardness reduction is from the VERTEX COVER problem, where vertices correspond to candidates, and edges correspond to votes so that each vote approves exactly the candidates corresponding to its two endpoints. It is easy to see that there is a vertex cover of cardinality κ if and only if there is a κ -committee so that the Hamming distance between this committee and each vote is at most κ . The \bar{k} -hardness of MAV-MULTIWINNER follows from that VERTEX COVER is W[1]-hard with respect to $m - \kappa$ where m is the number of vertices (candidates) [26].

Corollary 1 ([26, 50]) *MAV-MULTIWINNER is W[1]-hard with respect to \bar{k} even when every vote approves two candidates.*

It should be noted that VERTEX COVER with respect to $m - \kappa$ is exactly a parameterized variant of the INDEPENDENT SET problem. We provide the formal definition of the problem below because it will be used to establish our next intractability result.

An *independent set* of a graph is a subset of pairwise nonadjacent vertices.

κ -Independent Set (κ -IDS)

Input: A graph $G = (N, A)$ and an integer κ .
Parameter: κ .
Question: Does G admit an independent set of cardinality at least κ ?

Theorem 2 *CCAV-MULTIWINNER is W[1]-hard with respect to \bar{k} , even when every vote approves two candidates.*

Proof To prove Theorem 2, we offer a reduction from κ -IDS to CCAV-MULTIWINNER, similar to the one for MAV-MULTIWINNER discussed above by LeGrand [50]. Let $(G = (N, A), \kappa)$

be an instance of κ -IDS. We construct an instance $((C, V), k, d)$ of CCAV-MULTIWINNER as follows. For each vertex $c \in N$, we create one candidate denoted by the same symbol for simplicity. Let $C = N$ and let $m = |C|$. For each edge $\{c, c'\} \in A$, we create one vote v approving exactly c and c' , i.e., $v = \{c, c'\}$. Let V be the set of the created votes. Finally, we set $\bar{k} = \kappa$ (hence $k = m - \kappa$) and set $d = |A|$.

The correctness of the reduction is easy to check. If there is an independent set I of size \bar{k} , the CCAV score of the $(m - \bar{k})$ -committee $N \setminus I$ is d . Conversely, if there is an $(m - \bar{k})$ -committee w of CCAV score d , every vote has at least one of its approved candidates in the committee. Due to the construction, this implies that $N \setminus w$ is an independent set. □

For PAV, we also obtain a W[1]-hardness result via a reduction from the following problem.

Minimum (κ)-Vertex Subgraph (κ -MVS)

Input: A graph $G = (N, A)$ and two integers κ and ℓ .

Parameter: κ .

Question: Is there $S \subseteq N$ such that $|S| = \kappa$ and $G[N \setminus S]$ has at most ℓ edges?

Cai [17] proved that the κ -MVS problem is W[1]-hard even when the input graph is regular, i.e., all vertices have the same degree.

Theorem 3 *PAV-MULTIWINNER is W[1]-hard with respect to \bar{k} , even when every vote approves two candidates.*

Proof We provide a reduction from κ -MVS to PAV-MULTIWINNER. Let (G, κ, ℓ) be an instance of κ -MVS such that every vertex of G has degree r for some positive integer r . Let $G = (N, A)$ and let $m = |N|$. Without loss of generality, we assume that $\kappa < m$. We construct an instance $((C, V), k, d)$ of PAV-MULTIWINNER as follows. For each vertex $c \in N$, we create one candidate in C denoted by the same symbol, and for each edge $\{c, c'\} \in A$, we create one vote approving c and c' in V . We set $k = m - \kappa$, and hence $\bar{k} = \kappa$. Finally, we set $d = (m - \kappa) \cdot r - \frac{\ell}{2}$. It remains to show the correctness.

(\Rightarrow) Suppose that there exists $S \subseteq N$ so that $|S| = \kappa$ and $G[N \setminus S]$ contains exactly $\ell' \leq \ell$ edges. Then the PAV score of the committee $N \setminus S$ is

$$\frac{3}{2}\ell' + ((m - \kappa) \cdot r - 2\ell') = (m - \kappa) \cdot r - \frac{\ell'}{2} \geq d.$$

Obviously, $|N \setminus S| = m - |S| = k$. Therefore, the PAV-MULTIWINNER instance is a Yes-instance.

(\Leftarrow) Conversely, suppose that $w \subseteq C$ is an $(m - \kappa)$ -committee of PAV score at least d . Let ℓ' be the number of votes whose both approved candidates are in w . Due to the reduction, every candidate is approved by exactly r votes. Therefore, there are exactly $(m - \kappa) \cdot r - 2\ell'$ votes which have exactly one of their approved candidates in w . Hence, the PAV score of w is $\frac{3}{2}\ell' + ((m - \kappa) \cdot r - 2\ell')$, which is at least d . It immediately follows that $\ell' \leq \ell$, implying that $N \setminus w$ is a Yes-witness of the κ -MVS instance (G, κ, ℓ) . □

It should be pointed out that a reduction established by Skowron, Faliszewski, and Lang [67, Theorem 5] for showing NP-hardness of a related problem also implies that PAV-MULTIWINNER is $W[1]$ -hard with respect to \bar{k} .⁷

Now we move on to the parameter d . We observe that CCAV-MULTIWINNER is equivalent to the PARTIAL HITTING SET problem which has been intensively studied in the literature.

Partial Hitting Set

Input: A universe U , a collection \mathcal{S} of subsets of U , and two nonnegative integers a and b .
Parameter: b .
Question: Is there $S \subseteq U$ such that $|S| = a$ and S intersects at least b elements of \mathcal{S} ?

Clearly, by taking $C = U$, $V = \mathcal{S}$, $k = a$, and $d = b$ in the above definition we obtain CCAV-MULTIWINNER. Bläser [5] derived an algorithm running in time $O^*(2^{O(b)})$ for PARTIAL HITTING SET. This leads to the following corollary.

Corollary 2 ([5]) *CCAV-MULTIWINNER can be solved in time $O^*(2^{O(d)})$.*

Observe that every CCAV-MULTIWINNER instance where $d > n$ is a No-instance. As a result, Corollary 2 implies an algorithm for CCAV-MULTIWINNER running in time $O^*(2^{O(n)})$, which appreciably improves the $O^*(n^n)$ algorithm presented in [4].

Corollary 3 ([5]) *CCAV-MULTIWINNER can be solved in time $O^*(2^{O(n)})$.*

Bläser [5] also presented a randomized algorithm of running time $O^*((2e)^b)$ which solves the PARTIAL HITTING SET problem correctly with probability at least $1 - e^{-1}$, where e is the Napier's constant. We arrive at the following corollary.

Corollary 4 ([5]) *There is a randomized algorithm running in time $O^*((2e)^d)$ which solves CCAV-MULTIWINNER correctly with probability at least $1 - e^{-1}$.*

4.2 Parameters Δ_V and Δ_C

In this section, we study the two parameters Δ_V and Δ_C . First, as VERTEX COVER remains NP-hard when restricted to cubic graphs (i.e., 3-regular graphs) [38], the reduction for the NP-hardness of MAV-MULTIWINNER established by LeGrand [50] implies that MAV-MULTIWINNER is NP-hard even when every candidate is approved by three votes and every vote approves two candidates. It remains to study the cases where $\Delta_V \leq 1$ or $\Delta_C \leq 2$.

Recall that for a committee selection rule τ and an election $E = (C, V)$, $\tau(E, k)$ is the set of all optimal k -committees under τ . More precisely, for τ being MAV, $\tau(E, k)$ consists of all k -committees of C with the minimum Hamming distance to the votes, and

⁷ Their reduction is from VERTEX COVER restricted to 3-regular graphs, and a slight modification results in a reduction based on VERTEX COVER restricted to regular graphs which is $W[1]$ -hard parameterized by $m - \kappa$ [26].

for τ being CCAV and PAV, $\tau(E, k)$ consists of all k -committees of C with the maximum CCAV and PAV scores, respectively.

It is easy to see that every election $E = (C, V)$ where every vote approves only one candidate admits a k -committee, $k \leq |C|$, which is optimal with respect to MAV, CCAV, and PAV at the same time. In fact, any optimal k -committee under the approval voting (AV) is such a k -committee. The AV score of a candidate c is $|V(c)|$, the number of votes approving c , and AV selects k -committees with the maximum sum of AV scores of candidates in the committees.

Observation 1 Let $E = (C, V)$ be an election where every vote in V approves at most one candidate, and let $k \leq |C|$ be an integer. It holds that $AV(E, k) \subseteq MAV(E, k)$ and $AV(E, k) = CCAV(E, k) = PAV(E, k)$.

To see that $AV(E, k) \subseteq MAV(E, k)$ instead of $AV(E, k) = MAV(E, k)$ in Observation 1, consider an election with two candidates a and b , where a is approved by two votes, and b is approved by a third vote which does not approve a . For $k = 1$, both $\{a\}$ and $\{b\}$ are optimal with respect to MAV, but only $\{a\}$ is optimal with respect to AV.

The following corollary follows from Observation 1 and the clear fact that an optimal k -committee with respect to AV can be computed in polynomial time.

Corollary 5 For each $\tau \in \{MAV, CCAV, PAV\}$, τ -MULTIWINNER is polynomial-time solvable when $\Delta_V \leq 1$.

Regarding Δ_C , we have several polynomial-time solvability results for $\Delta_C \leq 2$. One of our results is based on the following polynomial-time solvable problem [52].

Simple b -Edge Cover of Multigraphs (SECM)

Input: A multigraph $G = (N, A)$, a function $f : N \rightarrow \mathbb{Z}^+$, and an integer κ .

Question: Is there $A' \subseteq A$ such that $|A'| \leq \kappa$ and every $v \in N$ is incident to at least $f(v)$ edges in A' ?

If we require $|A'| = \kappa$ in the above definition, we obtain the exact version of SECM (E-SECM). Clearly, E-SECM can also be solved in polynomial time.

Every election $E = (C, V)$ can be represented by a multihypergraph where every vote $v \in V$ is considered as a vertex and every candidate $c \in C$ is considered as an edge consisting of vertices in $V(c)$. When a candidate is approved by only one vote, there is a loop on this vote. We use $H(E)$ to denote this multihypergraph representing E . Clearly, given an election, its multihypergraph representation can be computed in polynomial time. When $\Delta_C \leq 2$, $H(E)$ degenerates to a multigraph.

For a class \mathcal{H} of multihypergraphs, we say that an election E is an \mathcal{H} -election if $H(E) \in \mathcal{H}$.

Theorem 4 MAV-MULTIWINNER and CCAV-MULTIWINNER are polynomial-time solvable if $\Delta_C \leq 2$.

Proof We derive polynomial-time algorithms for the special cases of MAV-MULTIWINNER and CCAV-MULTIWINNER stated in the theorem as follows. Let (E, k, d) be an instance of MAV-MULTIWINNER or CCAV-MULTIWINNER, where $E = (C, V)$. We first compute the

multigraph representation $H(E) = (V, A)$ of E which can be done in polynomial time, where A is the set of edges corresponding to candidates in C .

MAV. We solve MAV-MULTIWINNER by reducing it to E-SECM. Recall that for each $v \in V$, $|v|$ is the number of candidates approved by v which is equal to the number of edges incident to v in $H(E)$. Observe that the Hamming distance between every vote v and every k -committee can be at most $|v| + k$. Therefore, if $d \geq |v| + k$, we can safely remove v from V without changing the answer to the instance. In light of this fact, we assume now that $|v| + k > d$ for every $v \in V$. Let $f : V \rightarrow \mathbb{Z}^+$ be a function such that $f(v) = \lfloor \frac{|v|+k-d}{2} \rfloor$ for every $v \in V$. By setting $\kappa = k$ we complete the construction of an instance $(H(E), f, \kappa)$ of E-SECM. Assume that there exists $A' \subseteq A$ of cardinality κ so that every $v \in V$ is incident to at least $f(v)$ edges in A' . Let w be the k -committee corresponding to A' . The Hamming distance between every vote $v \in V$ and w is

$$|v \setminus w| + |w \setminus v| \leq (|v| - f(v)) + (k - f(v)) \leq d.$$

The proof for the other direction is analogous.

CCAV. We derive a greedy algorithm for CCAV-MULTIWINNER. Let H' be the graph obtained from $H(E)$ by

- (1) removing all loops, and
- (2) for every two vertices between which there are multiple edges, removing all but any arbitrary one of these multiple edges.

Let M be a maximum matching of H' , and let $V(M)$ be the set of vertices saturated by M . We distinguish between two cases.

- If $|M| \geq k$, we arbitrarily select k edges in M , and let w be the k -committee corresponding to these selected edges.
- If $|M| < k$, let w be the set of candidates corresponding to edges in M . Let E' be the election obtained from E by removing all votes in $V(M)$ and all candidates approved only by votes in $V(M)$. As M is a maximum matching of H' , no two votes of E' approve a common candidate. Then, for every nonempty vote in E' , we arbitrarily select one candidate approved by the vote. Let w' denote the set of all these selected candidates. If $|w'| \geq k - |M|$, we include into w any arbitrary $k - |M|$ candidates from w' ; otherwise, we include into w all candidates of w' together with any arbitrary $k - |w \cup w'|$ remaining candidates.

In either case, we conclude that the given instance of CCAV-MULTIWINNER is a Yes-instance if and only if $\text{CCAV}(V, w) \geq d$. \square

Now we study special cases of PAV-MULTIWINNER where Δ_C and Δ_V are very small integers. We need the following notions. A *path* is a graph comprised of a sequence v_1, v_2, \dots, v_t of t vertices and $t - 1$ edges so that there is an edge between two vertices if and only if they are consecutive in the sequence. A *cycle* is a graph obtained from a path by adding an edge between the first and the last vertices. A *hairstick* is a graph obtained from a path by adding one loop either on the first vertex or on the last vertex. A *double headed hairstick* (DH-hairstick) is a graph obtained from a path by adding one loop on both the first vertex and on the last vertex. We refer to Fig. 2 for an illustration of these graphs.

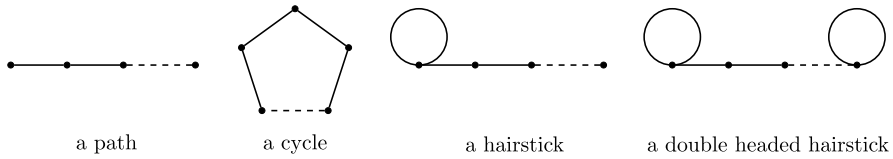


Fig. 2 An illustration of cycles, paths, hairsticks, and DH-hairsticks

Lemma 2 Let \mathcal{H} be the set of all paths, cycles, hairsticks, and DH-hairsticks. Then, given an \mathcal{H} -election, we can compute a PAV optimal k -committee of the election in polynomial time.

We defer the proof of Lemma 2 to the appendix.

Theorem 5 PAV-MULTIWINNER is polynomial-time solvable if $\Delta_C \leq 1$ or $\Delta_V = \Delta_C = 2$.

Proof Let $I = (E, k, d)$ be an instance of PAV-MULTIWINNER where $E = (C, V)$.

We consider first the case where $\Delta_C \leq 1$. The following algorithm finds an optimal k -committee. First, let $w = \emptyset$. Then, we arrange votes in V in a cyclic order (the relative orders of the votes do not matter), and starting from any arbitrary vote we consider the votes one-by-one in a clockwise order. In particular, if the currently considered vote v has at least one approved candidate not contained in w , i.e., $v \setminus w \neq \emptyset$, we add any arbitrary candidate from $v \setminus w$ into w ; otherwise we proceed to the next vote. The procedure runs until $|w| = k$ or w cannot be expanded in the way described above. Finally, we conclude that I is a Yes-instance if and only if $\text{PAV}(V, w) \geq d$.

Now we consider the case where $\Delta_C = \Delta_V = 2$. We first compute the multigraph representation $H(E)$ of E . Given a subgraph of $H(E)$, the subelection of E restricted to this subgraph refers to E restricted to candidates and votes corresponding respectively to the edges and the vertices of the subgraph. Observe that every connected component of $H(E)$ is either a path, a cycle, a hairstick, or a DH-hairstick, and by Lemma 2, for every integer j , an optimal j -committee of each subelection restricted to a connected component can be computed in polynomial time. Based on this, we derive a dynamic programming algorithm. Precisely, let (H_1, H_2, \dots, H_z) be an arbitrary order of the connected components of $H(E)$. For each $i \in [z]$, let $m^{\leq i}$ be the number of edges in the first i connected components in the order. We maintain a table $T(i, j)$ where $i \in [z]$ and $j \leq \min\{k, m^{\leq i}\}$ is a nonnegative integer. We define $T(i, j)$ as the PAV score of an optimal j -committee in the election restricted to the first i connected components. By Lemma 2, $T(i, j)$ for all possible j can be computed in polynomial time. We use the following recursion to compute $T(i, j)$, assuming that all entries $T(i', j')$ such that $i' < i$ have been computed. For each $j' \leq j$, let $d(j')$ be the PAV score of an optimal j' -committee of E restricted to H_i , which can be computed in polynomial time by Lemma 2. Then, we have that

$$T(i, j) = \max_{\substack{j' \in [j] \cup \{0\}, \\ j - j' \leq m^{\leq i-1}}} \{d(j') + T(i - 1, j - j')\}.$$

After all entries are computed, we conclude that the given instance I is a Yes-instance if and only if $T(z, k) \geq d$. □

5 Combined parameters

The study in the previous section revealed that the parameters k , \bar{k} , Δ_V , and Δ_C generally lead to fixed-parameter intractability results. In this section, we explore whether combining two or three of them offers us FPT-results.

5.1 Combining k with Δ_V and Δ_C

We first consider the combined parameter $k + \Delta_C$, starting with an FPT-result for MAV-MULTIWINNER.

Theorem 6 *MAV-MULTIWINNER is FPT with respect to $k + \Delta_C$.*

Proof We prove the theorem by giving an FPT-algorithm for MAV-MULTIWINNER as follows. Let $I = (E, k, d)$ be an instance of MAV-MULTIWINNER, where $E = (C, V)$. Let $n = |V|$. If $n \leq k \cdot \Delta_C + 1$, then as MAV-MULTIWINNER is FPT with respect to n [57], we can solve I in FPT-time in $k + \Delta_C$. Otherwise, let (v_1, v_2, \dots, v_n) be a linear order on V such that $|v_i| \geq |v_{i+1}|$ for every $i \in [n - 1]$, i.e., v_i approves at least the same number of candidates as v_{i+1} does. Then, the algorithm deletes the last $n - (k \cdot \Delta_C + 1)$ votes in this order, and solves the remaining instance by an FPT-algorithm with respect to the number of votes (e.g., the one presented in [57]).

It remains to prove the correctness of the algorithm. Let E' be the election after the deletion of the votes as described above. Clearly, every k -committee of the original election with MAV score d is a k -committee of E' with MAV score at most d . To show the correctness for the opposite direction, let w be a k -committee of E' with MAV score d . As every candidate is approved by at most Δ_C votes, at most $k \cdot \Delta_C$ votes in E' intersect w . As E' contains the first $k \cdot \Delta_C + 1$ votes in the order defined above, there exists $i \in [k \cdot \Delta_C + 1]$ so that the vote v_i does not approve any candidate from w . The Hamming distance between w and v_i is $k + |v_i| \leq d$. As $|v_j| \leq |v_i|$ for every deleted vote v_j , $j \geq k \cdot \Delta_C + 2$, the Hamming distance between w and v_j is at most $k + |v_j| \leq k + |v_i| \leq d$. It follows that w has MAV score d in the election E . \square

Let us move on to CCAV. Obviously, if w is a k -committee, then at most $k \cdot \Delta_C$ votes intersect w . Hence, the CCAV score of every optimal k -committee is bounded from above by $k \cdot \Delta_C$. This observation leads to a simple algorithm for CCAV-MULTIWINNER: if $d > k \cdot \Delta_C$, return “No”; otherwise, solve it by Corollary 2 in time $O^*(2^{O(d)}) = O^*(2^{O(k \cdot \Delta_C)})$.

Corollary 6 *CCAV-MULTIWINNER is FPT with respect to $k + \Delta_C$.*

Now we study the combined parameter $k + \Delta_V$. Observe that MAV-MULTIWINNER admits a straightforward FPT-algorithm with respect to $k + \Delta_V$ based on an algorithm in [57]: if $d > k + \Delta_V$, every k -committee has MAV score at most d , and hence we directly conclude that the given instance is a Yes-instance; otherwise, we solve it by the FPT-algorithm with respect to d proposed in [57].

Corollary 7 *MAV-MULTIWINNER is FPT with respect to $k + \Delta_V$.*

Aziz et al. [3] proved that PAV-MULTIWINNER is $W[1]$ -hard when the parameter is k even if $\Delta_V = 2$. This implies that PAV-MULTIWINNER is $W[1]$ -hard with respect to $k + \Delta_V$. For CCAV, Betzler, Slinko, and Uhlmann [4] proved that CCAV-MULTIWINNER is $W[2]$ -hard with respect to k by a reduction from the HITTING SET problem. However, in the reduction the maximum number of candidates approved by a vote is not bounded from above by a constant. Using a reduction from the PARTIAL VERTEX COVER problem, we show that CCAV-MULTIWINNER is $W[1]$ -hard with respect to k in this special case.

Partial Vertex Cover (PVC)

Input: A graph G and two integers κ and ℓ .
Parameter: κ .
Question: Is there a subset $S \subseteq V(G)$ such that $|S| = \kappa$ and S covers at least ℓ edges of G ?

It is known that PVC is $W[1]$ -hard with respect to κ [41].

Theorem 7 CCAV-MULTIWINNER is $W[1]$ -hard with respect to k even if every vote approves two candidates.

Proof Given an instance (G, κ, ℓ) of PVC, we create a CCAV-MULTIWINNER instance as follows. We regard each vertex as a candidate and regard each edge as a vote approving exactly the two candidates corresponding to its two endpoints. The reduction is completed by setting $k = \kappa$ and $d = \ell$. The correctness is easy to see. \square

5.2 Combining \bar{k} with Δ_V and Δ_C

We have shown that MAV-MULTIWINNER, CCAV-MULTIWINNER, and PAV-MULTIWINNER are $W[1]$ -hard or $W[2]$ -hard with respect to the single parameter \bar{k} even when $\Delta_V = 2$. It follows that these problems are $W[1]$ -hard or $W[2]$ -hard when parameterized by $\bar{k} + \Delta_V$. Hence, we focus only on the combined parameter $\bar{k} + \Delta_C$. We first prove that MAV-MULTIWINNER is FPT with respect to this parameter by reducing it to an FPT problem which is a generalization of the r -SET PACKING problem.

Generalized r -Set Packing (GrSP)

Input: A universe U , a multiset \mathcal{S} of r -subsets of U , a function $f : U \rightarrow \mathbb{N}_0$, and an integer κ .
Parameter: $\kappa + r$.
Question: Is there an $S \subseteq \mathcal{S}$ such that $|S| = \kappa$ and every $u \in U$ occurs in at most $f(u)$ elements of S ?

It is known that GrSP is FPT [79]⁸. It is easy to verify that the variant of the GrSP problem where each $s \in \mathcal{S}$ is of cardinality at most r , instead of exactly r , is reducible

⁸ The definition of GrSP in [79] requests $f(u)$ to be positive for all $u \in U$. However, it is fairly easy to see that if we allow $f(u) = 0$ for some $u \in U$, we can safely remove all $X \in \mathcal{S}$ such that $u \in X$ from the instance without changing the answer to the instance. So, allowing $f(u) = 0$ for $u \in U$ does not destroy the fixed-parameter tractability of the problem.

to $GrSP$ in polynomial time: for each $s \in \mathcal{S}$ such that $|s| < r$, create $r - |s|$ new elements in U , add them into s , and set $f(u) = 1$ for every newly introduced element in U . Therefore, the above variant is also FPT. In the following, we use $Gr_{\leq}SP$ to denote this variant.

Theorem 8 *MAV-MULTIWINNER is FPT with respect to $\bar{k} + \Delta_C$.*

Proof Let $I = (E, k, d)$ be an instance of MAV-MULTIWINNER, where $E = (C, V)$. Note that the Hamming distance between each vote v and each k -committee is at least $|k - |v||$. Therefore, if there exists a vote $v \in V$ such that $|v| < k$ and $d < k - |v|$, we immediately conclude that the given instance I is a No-instance. In what follows, we assume that for all $v \in V$ either it holds that $|v| \geq k$, or it holds that $|v| < k$ and $d \geq k - |v|$. We reduce I to an instance of $Gr_{\leq}SP$ in polynomial time as follows. Precisely, let $U = V$, and let $\mathcal{S} = \{V(c) : c \in C\}$. Clearly, each element of \mathcal{S} is of cardinality at most Δ_C . Hence, we set $r = \Delta_C$. Regarding the function f , for each $v \in V$ we define

$$f(v) = \left\lfloor \frac{d + |v| - k}{2} \right\rfloor.$$

By the above assumption, the function f is nonnegative. Finally, we define $\kappa = \bar{k} = |C| - k$. It remains to prove that the two instances are equivalent.

(\Rightarrow) Suppose that the $Gr_{\leq}SP$ instance is a Yes-instance. In particular, let $S \subseteq \mathcal{S}$ be such that $|S| = \kappa$ and every $u \in U$ occurs in at most $f(u)$ elements of S . Let C' be the subset of candidates corresponding to S . Clearly, $|C'| = |S| = \bar{k}$. We claim that $w = C \setminus C'$ is a k -committee with MAV score at most d . Due to the above construction, every vote v occurs in at most $f(v)$ submultisets from S . This implies that v has at least $|v| - f(v)$ of its approved candidates in w . Hence, the Hamming distance between v and w is at most

$$f(v) + (k - (|v| - f(v))) = 2 \cdot f(v) + k - |v| \leq d.$$

In other words, the instance I is a Yes-instance.

(\Leftarrow) Suppose that I is a Yes-instance, i.e., there is a k -committee $w \subseteq C$ such that $MAV(V, w) \leq d$. Let $C' = C \setminus w$. Then, for each vote $v \in V$ at least $\lceil \frac{|v| + k - d}{2} \rceil$ of its approved candidates must be in w . In other words, at most $|v| - \lceil \frac{|v| + k - d}{2} \rceil = f(v)$ of v 's approved candidates can be in $S = \{V(c) : c \in C'\}$. This implies that the $Gr_{\leq}SP$ instance is a Yes-instance. □

Next, we prove that CCAV-MULTIWINNER is FPT with respect to the combined parameter $\bar{k} + \Delta_C$ as well.

Theorem 9 *CCAV-MULTIWINNER is FPT with respect to $\bar{k} + \Delta_C$.*

Proof We prove Theorem 9 by deriving a branch-and-bound FPT-algorithm with respect to $\bar{k} + \Delta_C$. Let $I = ((C, V), k, d)$ be an instance of CCAV-MULTIWINNER. Throughout the algorithm, let $\bar{k} = |C| - k$.

We first remove from C all candidates not approved by any votes, and remove from V all empty votes. Then, we assume that neither of C and V is empty, since otherwise the instance can be solved trivially. Observe now that if a vote $v \in V$ approves more than \bar{k} candidates, any k -committee intersects v . In light of this observation, we need only to focus on votes approving at most \bar{k} candidates. Let $U \subseteq V$ be the submultiset of votes from V

approving at most \bar{k} candidates. Let $B = \bigcup_{v \in U} v$ be the set of candidates approved by votes from U . If $U = \emptyset$ or $|B| \leq k$, any k -committee containing B satisfies all votes. In this case, we conclude that I is a Yes-instance if and only if $d \leq |V|$. Assume now that $|B| \geq k + 1$. For each candidate $c \in C$, let $V^1(c)$ be the multiset of votes in U that approve c only, i.e., $V^1(c) = \{v \in U : v = \{c\}\}$. Let $c^* \in B$ be such that $|V^1(c^*)| \leq |V^1(c)|$ for all $c \in B$. Let $A = \bigcup_{v \in V(c) \cap U} v$. A significant observation is that there is an optimal k -committee which does not contain A . (A proof for the observation is given later.) In line with this observation, we solve the instance by branching on which candidate from A is not contained in a certain optimal k -committee. In particular, we create $|A|$ branching cases, one for each $x \in A$. In the branching case for $x \in A$, we reset $C := C \setminus \{x\}$ (note that this implies that \bar{k} is decreased by one), $d := d - |V \setminus U|$, and $V := U$, and solve the subinstance iteratively by the above procedure. We use two pruning criteria to determine when to terminate the branching: $d \leq 0$ or $\bar{k} = 0$. When we arrive at a branching node where $d \leq 0$ and $\bar{k} \geq 0$, it holds clearly that the given instance I is a Yes-instance, and thus in this case we terminate the whole algorithm by returning “Yes”. When we arrive at a branching node where $\bar{k} = 0$ and $d > 0$, we determine that I is a Yes-instance if $\text{CCAV}(V, C) \geq d$, and discard the corresponding branching case otherwise.

Now we prove the correctness of the above mentioned observation. To this end, assume that w is an optimal k -committee such that $A \subseteq w$. Then, we can obtain another optimal k -committee from w by replacing c^* (notice that $c^* \in A$) with any arbitrary candidate from $B \setminus w$ (as $|B| \geq k + 1$ such a candidate exists). The reason is that every vote from $(V(c^*) \cap U) \setminus V^1(c^*)$ approves at least one candidate from $w \setminus \{c^*\}$ (because $A \subseteq w$). Hence, removing c^* from w may only affect the existence of representatives of votes from $V^1(c^*)$ in the committee. However, as $|V^1(c^*)| \leq |V^1(c)|$ for all $c \in B$, adding any arbitrary candidate from $B \setminus w$ into w makes all votes from $V^1(c^*)$ which do not have any representatives before have representatives now.

Regarding the running time of the algorithm, note that $|A| \leq \Delta_C \cdot \bar{k}$. Hence, each branching node has at most $\Delta_C \cdot \bar{k}$ children, implying that the branching algorithm has running time $O^*((\Delta_C \cdot \bar{k})^{\bar{k}})$. \square

5.3 Combining d and Δ_V

It is known that MAV-MULTIWINNER and CCAV-MULTIWINNER are both FPT with respect to the parameter d . However, it is unknown whether PAV-MULTIWINNER is FPT with respect to d . Even though we are unable to resolve this open question, we provide an FPT-algorithm when combining d and Δ_V .

Theorem 10 *PAV-MULTIWINNER is FPT with respect to $d + \Delta_V$.*

Proof We prove the theorem by giving a branch-and-bound FPT-algorithm. Let $I = (E, k, d)$ be an instance of PAV-MULTIWINNER, where $E = (C, V)$.

First, if C contains a candidate approved by at least d votes in V , any k -committee including this candidate is a desired committee. So, in this case, we directly conclude that I is a Yes-instance.

Second, let $C_0 \subseteq C$ be the set of candidates not approved by any votes in V . We remove from the election (C, V) all candidates in C_0 , and reset $k := \min\{k, |C|\}$. It is easy to verify

that the two instances before and after this step are equivalent. Afterwards, if $k = |C|$, we conclude that I is a Yes-instance if and only if $\text{PAV}(V, C) \geq d$.

From now on, let us assume that every candidate is approved by at least one and at most $d - 1$ votes in V , and $k < |C|$. For each $S \subsetneq C$ and each $c \in C \setminus S$, we define

$$\Delta(S, c) = \text{PAV}(V, S \cup \{c\}) - \text{PAV}(V, S),$$

as the marginal contribution of c to the PAV score of the committee $S \cup \{c\}$. In our branching tree, each branching node is associated with a subset $S \subsetneq C$ of cardinality at most k , which is supposed to be contained in a desired committee. The root of the branching tree is associated with $S = \emptyset$. Suppose we are now at a branching node associated with a subset $S \subsetneq C$. Let $c \in C \setminus S$ be a candidate such that $\Delta(S, c) \geq \Delta(S, c')$ for all $c' \in C \setminus S$. Let $A = (\bigcup_{v \in V(c)} v) \setminus S$ be the set of candidates approved by votes approving c but not contained in S . Then, we create $|A|$ branching cases, one for each $x \in A$. The set associated with the branching case for $x \in A$ is $S \cup \{x\}$. The correctness of our branching is rooted in the fact (a proof supporting this fact is provided later) that, if the given instance I is a Yes-instance, there exists at least one desired k -committee which includes at least one candidate from A .

We terminate the branching when the branching depth of the current branching node reaches $\min\{k, d \cdot \Delta_V\}$. Recall that the branching depth of a branching node is the number of edges on the path from the root to the node in the branching tree. By our branching strategy, the branching depth of a branching node associated with S is exactly $|S|$. Therefore, when $k \leq d \cdot \Delta_V$ and the branching depth of the current node associated with S reaches k , we have that $|S| = k$. In this case, if S has PAV score at least d with respect to V , we conclude that the given instance I is a Yes-instance. Consider now the case where $k > d \cdot \Delta_V$. Note that $\Delta(S, x) \geq \frac{1}{\Delta_V}$ for each possible S and x . Therefore, if the current branching node associated with S has branching depth $d \cdot \Delta_V$, S is a k' -committee, $k' \leq k$, of PAV score at least d . As a result, when $k > d \cdot \Delta_V$ and the current node has branching depth $d \cdot \Delta_V$, we directly conclude that I is a Yes-instance. If none of the branching nodes leads to a “Yes”-answer, we conclude that I is a No-instance.

Note that $|A| \leq |V(c)| \cdot \Delta_V \leq d \cdot \Delta_V$, where A is as defined above. Therefore, the running time of the whole algorithm is bounded by $O^*((d \cdot \Delta_V)^{d \cdot \Delta_V})$.

To show the correctness, it suffices to prove the following claim (corresponding to the aforementioned fact).

Claim *Let $S \subsetneq C$ be a subset of at most $k - 1$ candidates, and let $c \in C \setminus S$ be a candidate such that $\Delta(S, c) \geq \Delta(S, c')$ for all $c' \in C \setminus S$. Let $A = (\bigcup_{v \in V(c)} v) \setminus S$. Then, if there is a k -committee $w \subsetneq C$ such that $\text{PAV}(V, w) \geq d$, $S \subsetneq w$, and $A \cap w = \emptyset$, there exists a k -committee $w' \subsetneq C$ such that $\text{PAV}(V, w') \geq d$, $S \subseteq (w \cap w')$, and $A \cap w' \neq \emptyset$.*

Now we prove the claim. Let S , c , and A be as stipulated in the claim. Suppose that there is a k -committee $w \subsetneq C$ such that $\text{PAV}(V, w) \geq d$, $S \subsetneq w$, and $A \cap w = \emptyset$. Let c' be any candidate in $w \setminus S$, and let $w' = w \setminus \{c'\} \cup \{c\}$. Obviously, $S \subseteq (w \cap w')$ and $A \cap w' \neq \emptyset$. To complete the proof, it suffices to show that $\text{PAV}(V, w') \geq d$. First, as $w \cap A = \emptyset$ and $S \subseteq w$, it holds that $\Delta(w, c) = \Delta(S, c)$. Second, as $S \subseteq w \setminus \{c'\}$, it holds that $\Delta(w \setminus \{c'\}, c') \leq \Delta(S, c')$. Third, it is clear that $\Delta(w \setminus \{c'\}, c) \geq \Delta(w, c)$. We also reiterate that $\Delta(S, c) \geq \Delta(S, c')$. Putting this all together, we obtain $\Delta(w \setminus \{c'\}, c) \geq \Delta(w \setminus \{c'\}, c')$ which implies $\text{PAV}(V, w') \geq \text{PAV}(V, w) \geq d$. The proof is completed. \square

We have derived many FPT-algorithms with respect to different single parameters and their combinations. Our results reveal that MAV-MULTIWINNER and CCAV-MULTIWINNER are FPT with respect to any combinations of three single parameters studied so far. Unfortunately, we have only a few FPT-results for PAV-MULTIWINNER even for combinations of two single parameters. We finish this section by remarking that PAV-MULTIWINNER is FPT with respect to $k + \Delta_V + \Delta_C$: if $d > k \cdot \Delta_C$, return “No”; otherwise, solve the instance in FPT-time via Theorem 10.

Corollary 8 *PAV-MULTIWINNER is FPT with respect to $k + \Delta_V + \Delta_C$.*

6 Structural parameters

In this section, we study two structural parameters of incidence graphs of elections. For an election E , let G_E denote the incidence graph of E .

6.1 Treewidth

Treewidth is a widely-studied notion to measure the closeness of a graph to a tree [63]. It has been shown that a great deal of graphs stemming from innumerable combinatorial problems in a variety of areas have bounded treewidth (see, e.g., [47, 56, 71, 74]). Additionally, it has been scrutinized that even when treewidth is large, using tree decompositions could also be helpful for designing algorithms [55]. From a theoretical point of view, a myriad of NP-hard problems are known to be FPT with respect to the parameter treewidth [21]. With regard to the applicability of treewidth in multiwinner voting, we point out that the treewidth of the incidence graph of an election is no greater than the number m of candidates and the number n of voters in the election, implying that any FPT-algorithm for τ -MULTIWINNER with respect to the treewidth also runs in FPT-time in m and n .

A *tree decomposition* of a graph $G = (N, A)$ is a tuple (T, \mathcal{B}) , where $T = (L, F)$ is a rooted tree with vertex set L and edge set F , and $\mathcal{B} = \{B_x \subseteq N : x \in L\}$ is a collection of subsets of vertices of G such that the following three conditions are fulfilled simultaneously:

- For each vertex $v \in N$ in G there exists at least one $B_x \in \mathcal{B}$ such that $v \in B_x$, i.e., every vertex of G is in at least one element of \mathcal{B} .
- For each edge $\{v, u\} \in A$ in G there exists at least one $B_x \in \mathcal{B}$ such that $v, u \in B_x$, i.e., every edge of G is contained in at least one element of \mathcal{B} .
- If a vertex $v \in N$ is in $B_x, B_y \in \mathcal{B}$, then v is in every $B_z \in \mathcal{B}$ where z is a vertex on the unique path between x and y in T .

The *width* of the tree decomposition is defined as $\max_{B \in \mathcal{B}} |B| - 1$. The *treewidth* of a graph G , denoted $\omega(G)$, is the minimum possible width of tree decompositions of G . Elements of \mathcal{B} are called *bags*. To avoid confusion, in the following we call vertices of T *nodes*.

A more refined notion commonly used in designing FPT-algorithms is the so-called nice tree decomposition [9]. In particular, a *nice tree decomposition* (T, \mathcal{B}) of a graph G is a tree decomposition of G which further satisfies the following conditions simultaneously:

- Every bag $B \in \mathcal{B}$ associated with the root or a leaf of T is empty, i.e., $B_x = \emptyset$ if x is the root or a leaf of T .
- Nonleaf nodes of T are categorized into *introduce nodes*, *forget nodes*, and *join nodes* such that:
 - Each introduce node x has exactly one child y such that $B_y \subsetneq B_x$ and $|B_x \setminus B_y| = 1$, i.e., B_x has exactly one more element than B_y .
 - Each forget node x has exactly one child y such that $B_x \subsetneq B_y$ and $|B_y \setminus B_x| = 1$, i.e., B_x is obtained from B_y by removing one element.
 - Each join node x has exactly two children y and z such that $B_x = B_y = B_z$.

It is easy to see from the definition that in a nice tree decomposition, each vertex can be introduced multiple times but can be only forgotten once. Moreover, when a vertex is introduced in a bag B_x , all of its neighbors contained in bags associated with nodes in the subtree rooted at x are contained in B_x .

Lemma 3 ([45]) *Let G be a graph of p vertices. Then, given a tree decomposition of G of width ω , a nice tree decomposition of G of width ω having $O(p \cdot \omega)$ nodes can be computed in polynomial time.*

It has long been known that calculating treewidth is NP-hard even for bipartite graphs [7]. However, determining whether the treewidth of a graph is at most ω is FPT with respect to ω and, moreover, powerful heuristic and approximation algorithms for calculating treewidth have been perpetually reported [6, 8, 72]. Besides, treewidth of chordal bipartite graphs can be computed in polynomial time,⁹ and the gap between chordal bipartite graphs and general bipartite graphs is quite small [46]. It is also well-known that every graph of p vertices admits an optimal tree decomposition of $O(p)$ nodes. Then, by Lemma 3, when we study FPT-algorithms with respect to treewidth or a combined parameter involving treewidth, it does not lose any generality to assume that a nice tree decomposition is given.

Theorem 11 *CCAV-MULTIWINNER can be solved in time $O^*(4^\omega)$ if a nice tree decomposition of width ω and of $\text{poly}(p)$ nodes of the incidence graph of the input election is given, where p is the number of vertices of the incidence graph.*

Proof Let (E, k, d) be an instance of CCAV-MULTIWINNER where $E = (C, V)$ and $k \leq |C|$. In addition, let (T, \mathcal{B}) be a nice tree decomposition of G_E of width ω that has $\text{poly}(p)$ nodes, where $p = |C| + |V|$. We design a dynamic programming algorithm running in time $O^*(4^\omega)$ as follows.

For each bag $B_x \in \mathcal{B}$ associated with a node x in the tree T , let $C(B_x)$ and $V(B_x)$ be the set of candidate-vertices and the set of vote-vertices contained in B_x , respectively. Moreover, let $C(T_x)$ be the set of candidates in bags associated with nodes in the subtree rooted at x . Obviously, $C(B_x) \subseteq C(T_x)$. We maintain for each node x in T a 3-dimensional table $D_x(C', V', k')$, where C' is a subset of $C(B_x)$ such that $|C'| \leq k$, V' is a submultiset of $V(B_x)$, and k' is an integer such that $|C'| \leq k' \leq \min\{k, |C(T_x)|\}$. We say that a k' -committee w

⁹ A chordal bipartite graph is a bipartite graph without induced cycles of length at least six.

is a valid committee for the entry $D_x(C', V', k')$ if all the following conditions hold simultaneously:

- (1) $w \subseteq C(T_x)$;
- (2) $C' = C(B_x) \cap w$; and
- (3) a vote in $V(B_x)$ has a representative in w if and only if this vote is in V' , i.e., for every $v \in V(B_x)$ it holds that $v \cap w \neq \emptyset$ if and only if $v \in V'$.

The value of the entry $D_x(C', V', k')$ is the CCAV score of an optimal valid k' -committee for the entry. If $D_x(C', V', k')$ admits no valid k' -committees, we define $D_x(C', V', k') = -\infty$. Clearly, each table associated with a bag has at most $2^{\omega+1} \cdot (k + 1)$ entries.

The algorithm updates the tables from those associated with the leaves up to the one associated with the root of T . The values of entries associated with leaves are all 0 (recall that each leaf bag is empty). We update the entry $D_x(C', V', k')$ associated with a nonleaf node x in T as follows.

First, if some candidate from C' is approved by some vote from $V(B_x) \setminus V'$, we set $D_x(C', V', k') = -\infty$. Otherwise, we consider the following cases with respect to the types of x .

- x is a join node
- Let y and z be the two children of x . We have $B_x = B_y = B_z$. In this case, we let

$$D_x(C', V', k') = \max_{\substack{k'_1 + k'_2 = k' - |C'| \\ 0 \leq k'_1 \leq |C(T_y) \setminus C'| \\ 0 \leq k'_2 \leq |C(T_z) \setminus C'| \\ V'_1, V'_2 \subseteq V', V'_1 \cup V'_2 = V'}} \{D_y(C', V'_1, k'_1 + |C'|) + D_z(C', V'_2, k'_2 + |C'|) - |V'_1 \cap V'_2|\}.$$

It is obvious that there are $O^*(2^\omega)$ different combinations of k'_1, k'_2, V'_1 , and V'_2 to consider in the max function. As a result, $D_x(C', V', k')$ can be computed in $O^*(2^\omega)$ time.

- x is an introduce node
- Let y be the child of x , and let $\{h\} = B_x \setminus B_y$. We further distinguish between two sub-cases.
 - h is a vote
 - If $h \notin V'$, then due to the above discussion h does not approve any candidates from C' . We set $D_x(C', V', k') = D_y(C', V', k')$. If $h \in V'$, then we set $D_x(C', V', k') = D_y(C', V' \setminus \{h\}, k') + 1$ if h approves at least one candidate from C' ; and set $D_x(C', V', k') = -\infty$ otherwise.
 - h is a candidate
 - If $h \notin C'$, we set $D_x(C', V', k') = -\infty$ when $k' = |C(T_x)|$, and set $D_x(C', V', k') = D_y(C', V', k')$ when $k' < |C(T_x)|$. If $h \in C'$, let $V_x(h)$ be the subset of votes in V' approving h . Then, we set

$$D_x(C', V', k') = \max_{U \subseteq V_x(h)} \{D_y(C' \setminus \{h\}, V' \setminus U, k' - 1) + |U|\}.$$

Note that in this case if $V_x(h) = \emptyset$, we have that $D_x(C', V', k') = D_y(C' \setminus \{h\}, V', k' - 1)$. As $|V_x(h)| \leq \omega$, $D_x(C', V', k')$ can be calculated in $O^*(2^\omega)$ -time.

- x is a forget node
- Let y be the child of x and let $\{h\} = B_y \setminus B_x$. We consider the following two subcases.
 - h is a vote
 - We set $D_x(C', V', k') = \max \{D_y(C', V', k'), D_y(C', V' \cup \{h\}, k')\}$.
 - h is a candidate
 - Clearly, $h \notin C'$. If $|C'| = k'$ or h is approved by some votes in $V(B_x) \setminus V'$, we set $D_x(C', V', k') = D_y(C', V', k')$; otherwise, let

$$D_x(C', V', k') = \max \{D_y(C', V', k'), D_y(C' \cup \{h\}, V', k')\}.$$

By the definition of the table, $D_r(\emptyset, \emptyset, k)$ is the CCAV score of an optimal valid k -committee for the entry, where r is the root of T (recall that the bag associated with the root is empty). By Conditions (1)–(3) given above, every k -committee of C is valid for $D_r(\emptyset, \emptyset, k)$. Therefore, after all tables are computed, we conclude that the given instance of CCAV-MULTIWINNER is a Yes-instance if and only if $D_r(\emptyset, \emptyset, k) \geq d$.

It remains to analyze the running time of the algorithm. Each node in the nice tree decomposition is associated with a table of $O^*(2^\omega)$ entries. Due to the above procedure, calculating an entry corresponding to a forget node takes polynomial time, and calculating an entry corresponding to a join or an introduce node takes $O^*(2^\omega)$ time. As the nice tree decomposition has polynomially many nodes, the running time of the algorithm is $O^*(4^\omega)$. □

Similar algorithms for PAV-MULTIWINNER and MAV-MULTIWINNER can be derived but with the running time being bounded by $O^*((k + 1)^\omega)$. The reason is that in these cases we need to maintain more information in order to solve the problem. For example, for PAV-MULTIWINNER, we need to maintain for each vote in V' the number of its approved candidates that are supposed to be in a desired committee, which requests a table of size at least $O^*((k + 1)^\omega)$.

Theorem 12 *PAV-MULTIWINNER and MAV-MULTIWINNER are FPT with respect to the combined parameter $k + \omega$, if a nice tree decomposition of width ω having $\text{poly}(p)$ nodes of the incidence graph of the input election is given, where p is the number of vertices of the incidence graph.*

Proof To prove the theorem, we derive FPT-algorithms for PAV-MULTIWINNER and MAV-MULTIWINNER with respect to $k + \omega$. Let $I = (E, k, d)$ be an instance of τ -MULTIWINNER where $\tau \in \{\text{PAV}, \text{MAV}\}$, $E = (C, V)$, and $k \leq |C|$. In addition, let $p = |C| + |V|$, and let (T, \mathcal{B}) be a nice tree decomposition of G_E of width ω that has $\text{poly}(p)$ nodes. The algorithms have the same skeleton as the one in the proof of Theorem 11 for CCAV-MULTIWINNER. More concretely, we maintain a table for each node of T , compute the tables in a bottom-up manner, and determine if the given instance I is a Yes-instance according to the table associated with the root. To delineate the algorithms, we need the following notations. For an integer i , we define $f(i) = \sum_{j=1}^i \frac{1}{j}$ if $i \geq 1$ and define $f(i) = 0$ otherwise. For a function $\varphi : A \rightarrow B$ and an element $a \in A$, φ_{-a} denotes φ restricted to the domain $A \setminus \{a\}$, i.e., $\varphi_{-a} : A \setminus \{a\} \rightarrow B$ is a function such that for all $a' \in A \setminus \{a\}$ it holds that $\varphi_{-a}(a') = \varphi(a')$. Additionally, let $C(B_x)$, $V(B_x)$, and $C(T_x)$ be defined as in the proof of Theorem 11. Furthermore, in parallel with $C(T_x)$, we let $V(T_x)$ denote the multiset of all votes contained in bags associated with nodes in the subtree rooted at x .

PAV. For each node x in T , we maintain a table $D_x(C', k', \mu)$ where $C' \subseteq C(B_x)$, $|C'| \leq k' \leq \min\{k, |C(T_x)|\}$, and $\mu : V(B_x) \rightarrow [k] \cup \{0\}$ is a function. We note that if $V(B_x) = \emptyset$, μ is an empty function. We say that a k' -committee $w \subseteq C(T_x)$ is valid for $D_x(C', k', \mu)$ if Conditions (1) and (2) listed in the proof of Theorem 11, and the following condition are satisfied simultaneously:

$$(D) |v \cap w| = \mu(v) \text{ for all } v \in V(B_x).$$

We define $D_x(C', k', \mu)$ as the maximum possible PAV score of k' -committees valid for $D_x(C', k', \mu)$ with respect to $V(T_x)$. More precisely,

$$D_x(C', k', \mu) := \max_{\substack{w \subseteq C(T_x), |w| = k' \\ w \text{ is valid for } D_x(C', k', \mu)}} \text{PAV}(V(T_x), w),$$

if $D_x(C', k', \mu)$ admits at least one valid k' -committee, and $D_x(C', k', \mu) = -\infty$ otherwise.

The tables for the leaves can be computed trivially according to the definition of the tables. We show how to update an entry $D_x(C', k', \mu)$ by distinguishing the types of the node x .

- x is a join node
- Let y and z be the two children of x . We let

$$D_x(C', k', \mu) = \max_{k'_1, k'_2, \mu_1, \mu_2} \{D_y(C', k'_1 + |C'|, \mu'_1) - g(\mu'_1) + D_z(C', k'_2 + |C'|, \mu'_2) - g(\mu'_2) + g(\mu)\},$$

where

- (a) k'_1 and k'_2 run over all integers so that $k'_1 + k'_2 = k' - |C'|$, $0 \leq k'_1 \leq |C(T_y) \setminus C'|$, and $0 \leq k'_2 \leq |C(T_z) \setminus C'|$;
- (b) μ_1 and μ_2 run over all functions from $V(B_x)$ to nonnegative integers so that for all $v \in V(B_x)$ it holds that $\mu_1(v) + \mu_2(v) = \mu(v) - |v \cap C'|$;
- (c) for each $i \in [2]$, μ'_i is a function from $V(B_x)$ to $[k] \cup \{0\}$ so that $\mu'_i(v) = \mu_i(v) + |v \cap C'|$ for all $v \in V(B_x)$; and
- (d) for each $\rho \in \{\mu, \mu'_1, \mu'_2\}$, $g(\rho) = \sum_{v \in V(B_x)} f(\rho(v))$.

By Condition (4), the number of different combinations of μ_1 and μ_2 in (b) is bounded from above by $O^*((k + 1)^{\omega+1}) \cdot O^*((k + 1)^{\omega+1}) = O^*(k^{2\omega})$. As a result, an entry in this case can be computed in time $O^*(k^{2\omega})$. (Note that by the above recursion, when $V(B_x) = \emptyset$ we have that $D_x(C', k', \mu) = \max_{k'_1, k'_2} \{D_y(C', k'_1 + |C'|, \mu) + D_z(C', k'_2 + |C'|, \mu)\}$.)

- x is an introduce node
- Let y be the child of x , and let $\{h\} = B_x \setminus B_y$. We further distinguish between the following two subcases.
 - h is a vote
 - In this case, we set

$$D_x(C', k', \mu) = \begin{cases} -\infty, & \text{if } \mu(h) \neq |h \cap C'| \\ D_y(C', k', \mu_{-h}) + f(\mu(h)), & \text{otherwise} \end{cases}$$

- h is a candidate
- If $h \notin C'$, we set $D_x(C', k', \mu) = -\infty$ when $k' = |C(T_x)|$, and set $D_x(C', k', \mu) = D_y(C', k', \mu)$ when $k' < |C(T_x)|$. If $h \in C'$, we set $D_x(C', k', \mu) = -\infty$ if there exists $v \in V(B_x)$ such that $h \in v$ and $\mu(v) = 0$, and set $D_x(C', k', \mu) = D_y(C' \setminus \{h\}, k' - 1, \mu') + \sum_{v \in V(B_x), h \in v} \frac{1}{\mu(v)}$ otherwise, where $\mu' : V(B_y) \rightarrow [k] \cup \{0\}$ is the function so that for all $v \in V(B_y)$ it holds that $\mu'(v) = \mu(v) - |v \cap \{h\}|$. (If $V(B_x) = \emptyset$, μ' is an empty function.)
- x is a forget node
- Let y be the child of x and let $\{h\} = B_y \setminus B_x$. Similar to the above case we distinguish between the following two subcases.
 - h is a vote We set $D_x(C', k', \mu) = \max_{\substack{\mu' : V(B_y) \rightarrow [k] \cup \{0\} \\ \mu'_{-h} = \mu}} D_y(C', k', \mu')$. Obviously, we have at most $(k + 1)^{\omega+1}$ different functions μ' to check, and hence such an entry $D_x(C', k', \mu)$ can be computed in $O^*(k^\omega)$ time.
 - h is a candidate
In this case, we set

$$D_x(C', k', \mu) = \begin{cases} D_y(C', k', \mu), & \text{if } |C'| = k' \\ \max\{D_y(C', k', \mu), D_y(C' \cup \{h\}, k', \mu)\}, & \text{otherwise} \end{cases}$$

After the table for the root r is computed, we conclude that the given instance I is a Yes-instance if and only if $D_r(\emptyset, k, \mu) \geq d$, where μ is an empty function (recall that the bag associated with the root is empty). This is because that by the definition of the table, $D_r(\emptyset, k, \mu)$ is the maximum possible PAV score of a valid k -committee for $D_x(\emptyset, k, \mu)$ with respect to $V(T_r) = V$.

MAV. For each node x in T , we maintain a table $D_x(C', k', \mu)$ whose components are defined as with the ones for PAV. The validity of a k' -committee for an entry is also defined the same. However, in this case each entry takes only binary values 1 and 0. The entry $D_x(C', k', \mu)$ is 1 if and only if there is at least one k' -committee w which is valid for the entry (i.e., a k' -committee satisfying Conditions (1)–(2) in the proof of Theorem 11, and Condition (4) given above) and, moreover, for each $v \in V(T_x) \setminus V(B_x)$, it holds that $|w \cap v| \geq \frac{k+|v|-d}{2}$. Observe that, by the definition of nice tree decomposition, none of the votes in $V(T_x) \setminus V(B_x)$ approves any candidates from $C \setminus C(T_x)$. Therefore, for every k -committee $w' \subseteq C$ such that $w' \cap C(T_x) = w$ and every $v \in V(T_x) \setminus V(B_x)$, it holds that $v \cap w = v \cap w'$, implying that the Hamming distance between v and w' is at most d if and only if $|v \cap w| \geq \frac{k+|v|-d}{2}$. The requirement $|w \cap v| \geq \frac{k+|v|-d}{2}$ ensures that as long as the given instance admits a Yes-witness $w' \subseteq C$ such that $w' \cap C(T_x) = w$, the existence of this Yes-witness is safely preserved in the table.

The tables for the leaves can be computed trivially according to their definitions. We show how to update an entry $D_x(C', k', \mu)$ by distinguishing the types of the node x .

- x is a join node
- Let y and z be the two children of x . In this case, we set $D_x(C', k', \mu) = 1$ if and only if there are two functions $\mu_1, \mu_2 : V(B_x) \rightarrow [k] \cup \{0\}$ and two nonnegative integers k'_1 and k'_2 such that

- (a) $k'_1 + k'_2 = k' - |C'|$;
- (b) for all $v \in V(B_x)$ it holds that $\mu_1(v) + \mu_2(v) = \mu(v) - |v \cap C'|$ and, moreover,
- (c) $D_y(C', k'_1 + |C'|, \mu'_1) = D_z(C', k'_2 + |C'|, \mu'_2) = 1$, where for each $i \in [2]$, μ'_i is the function from $V(B_x)$ to $[k] \cup \{0\}$ so that $\mu'_i(v) = \mu_i(v) + |v \cap C'|$ for all $v \in V(B_x)$.

It is obvious that there are $O^*(k^{O(\omega)})$ different combinations of k'_1, k'_2, μ_1 , and μ_2 to enumerate. As a result, it takes $O^*(k^{O(\omega)})$ time to compute $D_x(C', k', \mu)$. (Note that when $V(B_x) = \emptyset$, the above recursion indicates that $D_x(C', k', \mu) = 1$ if and only if there are k'_1 and k'_2 as above so that $D_y(C', k'_1 + |C'|, \mu) = D_z(C', k'_2 + |C'|, \mu) = 1$.)

- x is an introduce node
- Let y be the child of x , and let $\{h\} = B_x \setminus B_y$. We further distinguish between the following two subcases.
 - h is a vote
 - We set $D_x(C', k', \mu) = 1$ if and only if $\mu(h) = |C' \cap h|$ and $D_y(C', k', \mu_{-h}) = 1$.
 - h is a candidate
 - If $h \notin C'$, we set $D_x(C', k', \mu) = 0$ when $k' = |C(T_x)|$, and set $D_x(C', k', \mu) = D_y(C', k', \mu)$ when $k' < |C(T_x)|$. If $h \in C'$, we set $D_x(C', k', \mu) = 1$ if and only if $D_y(C' \setminus \{h\}, k' - 1, \mu') = 1$ where $\mu' : V(B_y) \rightarrow [k] \cup \{0\}$ is a function so that for all $v \in V(B_y)$ it holds that $\mu'(v) = \mu(v) - |v \cap \{h\}|$. (When $V(B_x) = \emptyset$, $D_x(C', k', \mu) = D_y(C' \setminus \{h\}, k' - 1, \mu)$.)
- x is a forget node
- Let y be the child of x and let $\{h\} = B_y \setminus B_x$.
 - h is a vote
 - We set $D_x(C', k', \mu) = 1$ if and only if
 - (1) $\mu(h) \geq \frac{k+|h|-d}{2}$, and
 - (2) there is a function $\mu' : V(B_y) \rightarrow [k] \cup \{0\}$ so that $\mu = \mu'_{-h}$ and $D_y(C', k', \mu') = 1$.

The first condition is to ensure that there is a k -committee which contains a valid k' -committee w for the entry and the Hamming distance between w and h is at most d .

- h is a candidate
- If $|C'| = k'$, we set $D_x(C', k', \mu) = D_y(C', k', \mu)$. Otherwise, we set $D_x(C', k', \mu) = 1$ if and only if $D_y(C', k', \mu) + D_y(C' \cup \{h\}, k', \mu) \geq 1$.

After the table for the root r is computed, we conclude that the given instance I is a Yes-instance if and only if $D_r(\emptyset, k, \mu) = 1$, where μ is an empty function. The reason for this is that by the definition of the table, it holds that $D_r(\emptyset, k, \mu) = 1$ if and only if there is at least one k -committee $w \subseteq C(T_r) = C$ which is valid for $D_r(\emptyset, k, \mu)$ and, moreover, for each $v \in V(T_r) \setminus V(B_r) = V$, it holds that $|w \cap v| \geq \frac{k+|v|-d}{2}$.

The running times of the algorithms for PAV-MULTIWINNER and MAV-MULTIWINNER are dominated by the total size of all tables which is bounded from above by $O^*(k^{O(\omega)})$. □

6.2 Maximum matching

It is well-known that for every bipartite graph G , the size of a maximum matching of G equals the size of a minimum vertex cover of G [54]. As the treewidth of a graph is bounded from above by the size of a minimum vertex cover of the graph [37], it follows from Theorem 11 that CCAV-MULTIWINNER is FPT with respect to $\alpha(G_E)$, where E is a given election. We show that MAV-MULTIWINNER and PAV-MULTIWINNER are also FPT with respect to this parameter. Observe that the size of a maximum matching of G_E can be at most $\min\{m, n\}$, where m is the number of candidates and n is the number of votes in the election E . Consequently, every FPT-algorithm with respect to $\alpha(G_E)$ also runs in FPT-time in n and m .

Theorem 13 *MAV-MULTIWINNER and PAV-MULTIWINNER are FPT with respect to $\alpha(G_E)$, where E is the election in the input.*

Proof Let $I = (E, k, d)$ be an instance of τ -MULTIWINNER, where $E = (C, V)$ and $\tau \in \{\text{MAV}, \text{PAV}\}$. Let G_E be the incidence graph of E , and let M be a maximum matching of G_E . Hence, $\alpha(G_E) = |M|$. For simplicity, we write α for $\alpha(G_E)$. Let $C(M)$ and $V(M)$ be the set of candidates and the multiset of votes saturated by M , respectively. Obviously, $|C(M)| = |V(M)| = \alpha$. We derive algorithms for MAV and PAV as follows.

MAV. We split I into at most 2^α subinstances, each of which takes I and a subset $C' \subseteq C(M)$ of size at most k as input, and asks whether there is a k -committee $w \subseteq C'$ of MAV score at most d such that $C' = C(M) \cap w$. It is easy to see that I is a Yes-instance if and only if at least one of the subinstances is a Yes-instance. We show how to solve each subinstance in polynomial time.

Let $I' = (I, C')$ be a subinstance. Let v be a vote in $V \setminus V(M)$. Assume that w is a k -committee such that $w \cap C(M) = C'$. As M is a maximum matching of G_E , no vote from $V \setminus V(M)$ approves any candidate from $C \setminus C(M)$. Consequently, the Hamming distance between v and w is $|v| + k - 2|v \cap C'|$. Hence, if there is a vote in $V \setminus V(M)$ such that $|v| + k - 2|v \cap C'| > d$, the subinstance is a No-instance. So, let us assume that this is not the case. Now the task is to identify a subset H of $k - |C'|$ candidates from $C \setminus C(M)$ such that the Hamming distance between every vote from $V(M)$ and the committee $H \cup C'$ is at most d , or equivalently, for every $v \in V(M)$, H contains at least $\frac{|v|+k-d}{2} - |v \cap C'|$ of v 's approved candidates in $C \setminus C(M)$. For each $v \in V(M)$, let $f(v) = \frac{|v|+k-d}{2} - |v \cap C'|$. We reduce the subinstance into an ILP with a bounded number of variables. Specifically, for each $U \subseteq V(M)$, let C_U be the set of candidates in $C \setminus C(M)$ approved by all votes in U but not approved by any vote in $V(M) \setminus U$, and let $m_U = |C_U|$. For each $U \subseteq V(M)$, we create one nonnegative integer variable x_U which indicates the number of candidates from C_U that are supposed to be in a desired committee. The constraints are as follows.

- First, for each variable x_U we have that $0 \leq x_U \leq m_U$.
- Second, as we seek $k - |C'|$ candidates from $C \setminus C(M)$, it holds that

$$\sum_{U \subseteq V(M)} x_U = k - |C'|.$$

- Finally, for every $v \in V(M)$, it holds that

$$\sum_{U \subseteq V(M), v \in U} x_U \geq f(v).$$

This inequality ensures that the Hamming distance between every vote $v \in V(M)$ and the desired committee is at most d .

It is known that ILP is FPT with respect to the number of variables [51]. As the above ILP has at most 2^α variables, the subinstance can be solved in FPT-time in α . As there are at most 2^α subinstances, the whole algorithm runs in FPT-time in α .

PAV. The algorithm is analogous to the above one for MAV. First, we split the given instance I into subinstances by enumerating all possible intersections C' of $C(M)$ and a desired k -committee. Then, every nonempty vote $v \in V \setminus V(M)$ such that $v \cap C' \neq \emptyset$ provides a PAV score $\sum_{i=1}^{|v \cap C'|} \frac{1}{i}$ to every k -committee whose intersection with $C(M)$ is exactly C' . Let

$$d' = \sum_{v \in V \setminus V(M), v \cap C' \neq \emptyset} \sum_{i=1}^{|v \cap C'|} \frac{1}{i}.$$

The question now is equivalent to solving an instance $((C, V(M)), C', k, d - d')$ of ANNOTATED PAV-MULTIWINNER, which can be done in FPT-time in $|V(M)| = |M| = \alpha$ (Theorem 1). \square

7 Conclusion

We have investigated the parameterized complexity of τ -MULTIWINNER for τ being the three prevalent approval-based k -committee selection rules MAV, CCAV, and PAV, aiming at providing plentiful fixed-parameter tractability results with respect to meaningful parameters. We studied many natural single parameters, their combinations, and two structural parameters of incidence graphs of elections, and obtained an almost complete landscape of the parameterized complexity of τ -MULTIWINNER for $\tau \in \{\text{MAV}, \text{CCAV}, \text{PAV}\}$ with respect to these parameters. For a summary of our concrete results, we refer to Table 1. It should be noted that many of our tractability results were obtained by reducing τ -MULTIWINNER to well-studied graph/set problems. As advocated by several researchers [30, 36], a remarkable advantage of using the reduction scheme is that we can automatically update our results with the state-of-the-art of these graph/set problems. In addition, though many reductions are trivial and direct, we deem that pointing out the connections between τ -MULTIWINNER and the graph/set problems benefits researchers from different communities.

Our exploration leaves several intriguing problems for future research. We select two open questions that we believe to be the most challenging.

Open Question 1 Is PAV-MULTIWINNER FPT with respect to d ?

Open Question 2 Is PAV-MULTIWINNER FPT with respect to the treewidth of the incidence graph of the given election?

Besides these open questions, improving the FPT-algorithms presented in the paper or investigating the kernelizations of FPT problems are also promising avenues for future research. We remark that Agrawal et al. [1] studied kernelizations of the PARTIAL HITTING SET problem. Their results imply that CCAV-MULTIWINNER admits a polynomial kernel with respect to $d + \Delta_V$. It is interesting to see if the same holds for PAV-MULTIWINNER. Regarding MAV, Misra, Nabeel, and Singh [57] have shown that MAV-MULTIWINNER is unlikely to admit any polynomial kernels with respect to $d + m$ and $n + k$, assuming standard complexity hypothesis. This means that MAV-MULTIWINNER is unlikely to admit any polynomial kernels with respect to smaller parameters (e.g., $d, n, d + \Delta_V$, etc.).

Finally, we would like to mention that recently a few researchers have examined a number of structural parameters based on real electoral data or data generated by promising models, largely motivated by previous theoretical works on the parameterized complexity of voting problems with respect to these parameters (see, e.g., [10, 69]). Although graphs of small treewidth arise from various research areas have been continually reported in the literature, to the best of our knowledge, analogous works on the treewidth of incidence graphs of elections have not been conducted heretofore. We hope that our algorithms presented in Sect. 6 would inspire such an investigation.

Appendix

This appendix is devoted to the proof of Lemma 2. In the following, when we remove a candidate from an election, we remove it from the candidate set and from all votes approving the candidate. **Lemma 2.** Let \mathcal{H} be the set of all paths, cycles, hairsticks, and DH-hairsticks. Then, given an \mathcal{H} -election, we can compute a PAV optimal k -committee of the election in polynomial time.

Proof Let $E = (C, V)$ be an election whose multihypergraph representation $H(E)$ is either a path, a cycle, a hairstick, or a DH-hairstick. Let k be a nonnegative integer such that $k \leq |C|$. Let $n = |V|$. Note that n is also the number of vertices in $H(E)$. We consider all the possible cases of $H(E)$ as follows.

Case 1: $H(E)$ is a path or a cycle. We first compute a maximum matching M of $H(E)$. If $|M| \geq k$, it is easy to see that the k -committee corresponding to M is optimal. Otherwise, we further distinguish between two subcases based on the parity of n . Let w be the committee corresponding to M . If n is even, we know that M is a perfect matching. Let M' be the set of edges of $H(E)$ without M . Obviously, M' is also a matching of $H(E)$. Let $C(M')$ be the set of candidates corresponding to M' . As M is a perfect matching, adding each $c \in C(M')$ into any committee containing w and excluding c increases the PAV score of the committee by exactly one. In light of this fact, we add any $k - |M|$ arbitrary candidates from $C(M')$ into w , and return w . If n is odd, there is exactly one vertex v in $H(E)$ not saturated by M . We put into w one arbitrary

- candidate approved by v . Then, we arbitrarily add $k - 1 - |M|$ candidates from the remaining candidates into w , and return w .
- Case 2: $H(E)$ is a hairstick. If $k = n$, we return the whole set of candidates. Otherwise (i.e., $k < n$), a key observation is that there exist optimal k -committees which do not contain the candidate corresponding to the loop in $H(E)$. By this observation, we directly remove the loop-candidate from the election. Now, the election admits a path representation, and we use the algorithm described in Case 1 to compute an optimal k -committee.
- Case 3: $H(E)$ is a DH-hairstick. If $k = n + 1$, we return the whole set of candidates. Otherwise, there exist optimal k -committees which contain at most one of the two candidates corresponding to the loops in $H(E)$. In this case, we arbitrarily select one loop, and remove the corresponding candidate from the election. Then, we arrive at a hairstick representation of the election. The algorithm given in Case 2 is applied to solve the problem.

It is easy to see that the above algorithms in all cases run in polynomial time. □

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 62172446, and in part by the Open Project of Xiangjiang Laboratory under Grants 22XJ02002 and 22XJ03005. The authors would like to thank all anonymous reviewers who have provided helpful comments on the paper.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Agrawal, A., Choudhary, P., Jain, P., Kanesh, L., Sahlot, V., & Saurabh, S. (2018). Hitting and covering partially. In *COCOON* (pp. 751–763).
2. Aziz, H., Brill, M., Conitzer, V., Elkind, E., Freeman, R., & Walsh, T. (2017). Justified representation in approval-based committee voting. *Social Choice and Welfare*, 48(2), 461–485.
3. Aziz, H., Gaspers, S., Gudmundsson, J., Mackenzie, S., Mattei, N., & Walsh, T. (2015). Computational aspects of multi-winner approval voting. In *AAMAS* (pp. 107–115).
4. Betzler, N., Slinko, A., & Uhlmann, J. (2013). On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47, 475–519.
5. Bläser, M. (2003). Computing small partial coverings. *Information Processing Letters*, 85(6), 327–331.
6. Bodlaender, H. L. (2012). Fixed-parameter tractability of treewidth and pathwidth. In *The Multivariate Algorithmic Revolution and Beyond* (pp. 196–227).
7. Bodlaender, H. L. (1993). A tourist guide through treewidth. *Acta Cybernetica*, 11(1–2), 1–21.

8. Bodlaender, H. L., Drange, P. G., Dregi, M. S., Fomin, F. V., Lokshantov, D., & Pilipczuk, M. (2016). A $c^{\delta n}$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2), 317–378.
9. Bodlaender, H. L., & Kloks, T. (1991). Better algorithms for the pathwidth and treewidth of graphs. In *ICALP* (pp. 544–555).
10. Boehmer, N., & Schaar, N. (2023). Collecting, classifying, analyzing, and using real-world ranking data. In *AAMAS* (pp. 1706–1715).
11. Brams, S. J., Kilgour, D. M., & Sanver, M. R. (2007). A minimax procedure for electing committees. *Public Choice*, 132(3–4), 401–420.
12. Brandstädt, A., Le, V. B., & Spinrad, J. P. (1999). Graph classes: A survey. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, Philadelphia.
13. Brederbeck, R., Faliszewski, P., Kaczmarczyk, A., Knop, D., & Niedermeier, R. (2021). Parameterized algorithms for finding a collective set of items. In *AAAI* (pp. 1838–1845).
14. Brederbeck, R., Faliszewski, P., Kaczmarczyk, A., Niedermeier, R., Skowron, P., & Talmon, N. (2021). Robustness among multiwinner voting rules. *Artificial Intelligence*, 290, 103403.
15. Brederbeck, R., Faliszewski, P., Niedermeier, R., Skowron, P., & Talmon, N. (2020). Mixed integer programming with convex/concave constraints: Fixed-parameter tractability and applications to multi-covering and voting. *Theoretical Computer Science*, 814, 86–105.
16. Byrka, J., & Sornat, K. (2014). PTAS for minimax approval voting. In *WINE* (pp. 203–217).
17. Cai, L. (2008). Parameterized complexity of cardinality constrained optimization problems. *The Computer Journal*, 51(1), 102–121.
18. Chamberlin, J. R., & Courant, P. N. (1983). Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, 77(3), 718–733.
19. Clearwater, A., Puppe, C., & Slinko, A. (2015). Generalizing the single-crossing property on lines and trees to intermediate preferences on median graphs. In *IJCAI* (pp. 32–38).
20. Cornaz, D., Galand, L., & Spanjaard, O. (2012). Bounded single-peaked width and proportional representation. In *ECAI* (pp. 270–275).
21. Courcelle, B., & Mosbah, M. (1993). Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109(1–2), 49–82.
22. Cygan, M., Fomin, F. V., Kowalik, L., Lokshantov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., & Saurabh, S. (2015). *Parameterized algorithms*. Springer.
23. Cygan, M., Kowalik, L., Socala, A., & Sornat, K. (2018). Approximation and parameterized complexity of minimax approval voting. *Journal of Artificial Intelligence Research*, 63, 495–513.
24. Dom, M., Lokshantov, D., & Saurabh, S. (2014). Kernelization lower bounds through colors and IDs. *ACM Transactions on Algorithms*, 11(2), 13.
25. Downey, R. G., & Fellows, M. R. (2013). Fundamentals of parameterized complexity. Texts in computer science, Springer.
26. Downey, R. G., & Fellows, M. R. (1995). Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science*, 141(1–2), 109–131.
27. Downey, R. G., & Fellows, M. R. (1992). Fixed parameter tractability and completeness. In *Complexity Theory: Current Research* (pp. 191–225).
28. Elkind, E., Faliszewski, P., Skowron, P., & Slinko, A. (2017). Properties of multiwinner voting rules. *Social Choice and Welfare*, 48(3), 599–632.
29. Elkind, E., & Lackner, M. (2015). Structure in dichotomous preferences. In *IJCAI* (pp. 2019–2025).
30. Elkind, E., & Lackner, M. (2014). On detecting nearly structured preference profiles. In *AAAI* (pp. 661–667).
31. Endriss, U. (Ed.). (2017). *Trends in computational social choice*. AI Access.
32. Faliszewski, P., Skowron, P., Slinko, A., & Talmon, N. (2018). Multiwinner analogues of the plurality rule: Axiomatic and algorithmic perspectives. *Social Choice and Welfare*, 51(3), 513–550.
33. Faliszewski, P., Skowron, P., Slinko, A., & Talmon, N. (2016). Committee scoring rules: Axiomatic classification and hierarchy. In *IJCAI* (pp. 250–256).
34. Faliszewski, P., Skowron, P., & Talmon, N. (2017). Bribery as a measure of candidate success: Complexity results for approval-based multiwinner rules. In *AAMAS* (pp. 6–14).
35. Faliszewski, P., Slinko, A., Stahl, K., & Talmon, N. (2018). Achieving fully proportional representation by clustering voters. *Journal of Heuristics*, 24(5), 725–756.
36. Fernau, H., Fomin, F. V., Lokshantov, D., Mnich, M., Philip, G., & Saurabh, S. (2014). Social choice meets graph drawing: How to get subexponential time algorithms for ranking and drawing problems. *Tsinghua Science and Technology*, 19(4), 374–386.
37. Fomin, F. V., Liedloff, M., Montealegre, P., & Todinca, I. (2018). Algorithms parameterized by vertex cover and modular width, through potential maximal cliques. *Algorithmica*, 80(4), 1146–1169.

38. Garey, M. R., Johnson, D. S., & Stockmeyer, L. (1976). Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3), 237–267.
39. Gawron, G., & Faliszewski, P. (2022). Using multiwinner voting to search for movies. In *EUMAS* (pp. 134–151).
40. Ghosh, S., Mundhe, M., Hernandez, K., & Sen, S. (1999). Voting for movies: The anatomy of a recommender system. In *AGENTS* (pp. 434–435).
41. Guo, J., Niedermeier, R., & Wernicke, S. (2005). Parameterized complexity of generalized vertex cover problems. In *WADS* (pp. 36–48).
42. Gupta, S., Jain, P., Saurabh, S., & Talmon, N. (2021). Even more effort towards improved bounds and fixed-parameter tractability for multiwinner rules. In *IJCAI* (pp. 217–223).
43. Kilgour, D. M. (2010). Approval balloting for multi-winner elections. In J.-F. Laslier & M. R. Sanver (Eds.), *Handbook on Approval Voting* (pp. 105–124). Springer.
44. Kilgour, D. M., & Marshall, E. (2012). Approval balloting for fixed-size committees. In D. S. Felsenthal & M. Machover (Eds.), *Electoral systems-paradoxes, assumptions, and procedures* (pp. 305–326). Springer.
45. Kloks, T. (eds.) (1994). Treewidth: Computations and approximations. Lecture Notes in Computer Science, vol. 842. Springer.
46. Kloks, T., & Kratsch, D. (1995). Treewidth of chordal bipartite graphs. *Journal of Algorithms*, 19(2), 266–281.
47. Krause, P. K., Larisch, L., & Salfelder, F. (2020). The tree-width of C . *Discrete Applied Mathematics*, 278, 136–152.
48. Lackner, M., & Skowron, P. (2023). *Multi-winner voting with approval preferences*. Springer.
49. Lackner, M., & Skowron, P. (2021). Consistent approval-based multi-winner rules. *Journal of Economic Theory*, 192, 105173.
50. LeGrand, R. (2004). Analysis of the minimax procedure. Technical Report, Department of Computer Science and Engineering, Washington University, St. Louis, USA
51. Lenstra, H. W. (1983). Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4), 538–548.
52. Lin, A.P. (2011). The complexity of manipulating k -approval elections. In *ICAART* (Vol. 2, pp. 212–218). [arXiv:1005.4159](https://arxiv.org/abs/1005.4159).
53. Liu, H., & Guo, J. (2016). Parameterized complexity of winner determination in minimax committee elections. In *AAMAS* (pp. 341–349).
54. Lovász, L., & Plummer, M. D. (1986). *Matching theory*. Elsevier.
55. Maniu, S., Senellart, P., & Jog, S. (2019). An experimental study of the treewidth of real-world graph data. In *ICDT* (p. 12).
56. Marchand, B., Ponty, Y., & Bultheau, L. (2022). Tree diet: reducing the treewidth to unlock FPT algorithms in RNA bioinformatics. *Algorithms for Molecular Biology*, 17(1), 8.
57. Misra, N., Nabeel, A., & Singh, H. (2015). On the parameterized complexity of minimax approval voting. In *AAMAS* (pp. 97–105).
58. Peters, D. (2018). Single-peakedness and total unimodularity: New polynomial-time algorithms for multi-winner elections. In *AAAI* (pp. 1169–1176).
59. Peters, D., & Elkind, E. (2016). Preferences single-peaked on nice trees. In *AAAI* (pp. 594–600).
60. Peters, D., & Lackner, M. (2020). Preferences single-peaked on a circle. *Journal of Artificial Intelligence Research*, 68, 463–502.
61. Procaccia, A. D., Rosenschein, J. S., & Zohar, A. (2008). On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3), 353–362.
62. Procaccia, A. D., Rosenschein, J. S., & Zohar, A. (2007). Multi-winner elections: Complexity of manipulation, control and winner-determination. In *IJCAI* (pp. 1476–1481).
63. Robertson, N., & Seymour, P. D. (1986). Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3), 309–322.
64. Schrijver, A. (1986). *Theory of linear and integer programming*. Wiley.
65. Skowron, P. (2017). FPT approximation schemes for maximizing submodular functions. *Information and Computation*, 257, 65–78.
66. Skowron, P., & Faliszewski, P. (2017). Chamberlin-Courant rule with approval ballots: Approximating the maxcover problem with bounded frequencies in FPT time. *Journal of Artificial Intelligence Research*, 60, 687–716.
67. Skowron, P., Faliszewski, P., & Lang, J. (2016). Finding a collective set of items: From proportional multirepresentation to group recommendation. *Artificial Intelligence*, 241, 191–216.
68. Skowron, P., Yu, L., Faliszewski, P., & Elkind, E. (2015). The complexity of fully proportional representation for single-crossing electorates. *Theoretical Computer Science*, 569, 43–57.

69. Sui, X., Francois-Nienaber, A., & Boutilier, C. (2013). Multi-dimensional single-peaked consistency and its approximations. In *IJCAI* (pp. 375–382).
70. Thiele, T. N. (1895). Om Flerfoldsvalg. In: *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandling* (pp. 415–441).
71. Thorup, M. (1998). All structured programs have small tree-width and good register allocation. *Information and Computation*, 142(2), 159–181.
72. van der Zanden, T. C. (2019). Theory and practical applications of treewidth. Ph.D. thesis, Utrecht University, Netherlands.
73. West, D. B. (2000). *Introduction to graph theory*. Prentice-Hall.
74. Yamaguchi, A., Aoki, K. F., & Mamitsuka, H. (2003). Graph complexity of chemical compounds in biological pathways. *Genome Informatics*, 14, 376–377.
75. Yang, Y. (2021). A model of winners allocation. In: *AAAI* (pp. 5760–5767).
76. Yang, Y. (2020). On the complexity of destructive bribery in approval-based multi-winner voting. In *AAMAS* (pp. 1584–1592).
77. Yang, Y. (2019). Complexity of manipulating and controlling approval-based multiwinner voting. In *IJCAI* (pp. 637–643).
78. Yang, Y. (2019). On the tree representations of dichotomous preferences. In *IJCAI* (pp. 644–650).
79. Yang, Y., & Guo, J. (2017). The control complexity of r -approval: From the single-peaked case to the general case. *Journal of Computer and System Sciences*, 89, 432–449.
80. Yang, Y., & Wang, J. (2018). Parameterized complexity of multi-winner determination: More effort towards fixed-parameter tractability. In *AAMAS* (pp. 2142–2144).
81. Yu, L., Chan, H., & Elkind, E. (2013). Multiwinner elections under preferences that are single-peaked on a tree. In *IJCAI* (pp. 425–431).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.