Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

Efficient Algorithms for Electing Successive Committees

Pallavi Jain[1] and **Andrzej Kaczmarczyk**[2]
[1] Indian Institute of Technology, Jodhpur, India
[2] Department of Computer Science, The University of Chicago, Chicago, United States
pallavi@iitj.ac.in, akaczmarczyk@uchicago.edu

Abstract

In a recently introduced model of *successive committee elections*, for a given set of ordinal or approval preferences one aims to find a sequence of a given length of "best" same-size committees such that each candidate is a member of a limited number of consecutive committees. However, the practical usability of this model remains limited, as the described task turns out to be NP-hard for most selection criteria already for seeking committees of size three. Non-trivial or somewhat efficient algorithms for these cases are lacking too. Motivated by a desire to unlock the full potential of the described temporal model of committee elections, we devise (parameterized) algorithms that effectively solve the mentioned hard cases in realistic scenarios of a moderate number of candidates or of a limited time horizon.

1 Introduction

A non-profit organization (NPO) offers a 3-day personal development workshop for teenagers in a remote location. Featured activities include discussion sessions with three expert counselors who tackle the participants' questions in various topics of developing self awareness. Potential counselors agreed to participate in such discussions for at most two consecutive days to avoid excessive traveling and fatigue. Every counselor specializes in a limited selection of topics regarding self-awareness. The NPO wants to select three groups of three counselors, one group per day, to offer the participants as broad experience as possible. Hence, the selected groups must obey the counselors' consecutiveness requirement and also guarantee a diverse selection of topics covered each day by the respective counselor pair.

The NPO's task seemingly can be modeled as a *multi-winner voting* task [Faliszewski *et al.*, 2017; Lackner and Skowron, 2023]. Identifying counselors as candidates and each subarea as a voter, who approves for the candidates representing counselor's expertise, we want to select a diverse group of candidates. However, the classical model neglects the temporal aspect of the problem and thus misaligns with the task. Precisely, it fails to group the counselors in teams of three to serve on the three days of the workshop while adhering to the consecutiveness requirement.

The described shortcoming has recently been addressed by Bredereck *et al.* [2020], who proposed a suitable framework of *successive committee elections*. Here, based on a collection of votes, one selects a collection of multiple, ordered, same-size groups, called *committees*, of "best" candidates. In line with our toy example, each candidate in the selected committees must be a member of a single contiguous block of at most a given number of committees. Besides introducing the new model, the authors have studied associated problems through algorithmic lens. They studied four successive committee rules focused on maintaining the diversity of the chosen committees based on the Chamberlin–Courant rule [Chamberlin and Courant, 1983]. Further, they considered extensions of the widespread approval voting rule and weakly-separable scoring rules. Their study identified cases solvable in polynomial time mostly related to finding series of committees that consist of two candidates. While they showed that with very few exceptions the same task for committees of size three or more is computationally hard, they abstained from providing algorithms dealing with such cases.

Motivated by the desire of providing means of computing successive committees for criteria considered by Bredereck *et al.* [2020], we sidestep their hardness results by applying advanced algorithmic methods from parameterized complexity theory. We analyze the same set of rules they did and contribute several positive algorithmic results, as shown in Table 1. Consequently, (1) we provide the first algorithms that deal with the hard scenarios for committee size bigger than two, including cases focusing on diverse committees similar to our introductory example, and (2) our methods are furthermore computationally efficient (in the parameterized sense) for cases with a small number of candidates or committees to be selected. The mentioned cases of interest seem quite likely in practice. Indeed, planning too far ahead usually bears unacceptable risks (like unpredictable changes of voters preferences or dropping out of candidates). Too big pool of candidates is often undesired due to human perception limitation and is usually avoided by shortlisting (in a broad sense, e.g., by requiring petition signatures, deliberation processes, organizing pre-selection). As our algorithmic contributions extend the applicability of successive committee elections, we directly respond to the call of Boehmer and

Niedermeier [2021] to consider different paths of tractability of new models[1] better capturing *the changing nature of real-world problems*.

Some other models incorporate the time aspect into classical committee elections additionally allowing votes to change their vote over time. Bredereck *et al.* [2022] introduce such a model and analyze the (parameterized) computational complexity of several related questions. Their results, complemented with those of Kellerhals *et al.* [2021], offer a comprehensive computational landscape. Importantly, their model does not generalize the one we consider, as their requirements for the outcome on a series of committees are focused on (dis)similarity of neighboring committees. Deltl *et al.* [2023] deepen the study of the model of Bredereck *et al.* [2022] by studying new questions related mostly to the fairness of the outcomes towards the voters. The dynamic nature of preferences considered by the three listed works makes approaches therein impossible to adapt to our problems. In the light of the mentioned literature, our work takes a significant step towards addressing the somewhat neglected algorithmic study of successive committee elections.

Another related scenario of time-dependent voting where the task is to select a single candidate for each time step can be seen as a series of committees of size one. The literature on this model mostly focuses on proportionality and fairness from axiomatic [Lackner, 2020; Lackner and Maly, 2021; Chandak *et al.*, 2024], algorithmic [Elkind *et al.*, 2024b; Bulteau *et al.*, 2021] and experimental perspectives [Bulteau *et al.*, 2021; Chandak *et al.*, 2024]. See a survey by Elkind *et al.* [2024a] for more details on this topic.

2 Preliminaries

For a positive integer x, we use $[x]$ to denote set $\{1, 2, \dots, x\}$. An *election* $E = (C, V)$ consists of m candidates $C = \{c_1, c_2, \dots, c_m\}$ and a collection $V = \{v_1, v_2, \dots, v_n\}$ of n voters. We study *approval* and *ordinal* preferences. In the former preference type, we associate a voter $v_i \in V$ with their *approval set* $A(i)$ of candidates that v_i approves. In the ordinal preferences model each voter v_i ranks all candidates and so is identified with their (total and strict) *preference order* $\succ_i$. We denote by $\mathrm{pos}_i(c)$ a position of candidate $c \in C$ in some voter v_i's ranking $\succ_i$. There are multiple mathematical ways of relating ordinal preferences to approval preferences.[2] However, they all require decisions somewhat arbitrary from the practical perspective.

Fixed-Parameter Tractability. We say a computational problem is fixed-parameter tractable for some parameter x (being a part of the input) if there is an (parameterized) algorithm solving every instance $\mathcal{I}$ in time $\mathcal{O}(f(x)|\mathcal{I}|^c)$ for some constant c. We call such an algorithm an FPT(x)-algorithm. Under standard computational complexity assumptions, fixed-parameter tractability for some parameter x is excluded when the problem is W[t]-hard, $t \in \mathbb{N}$ with respect to x or when the problem is NP-hard for a fixed value of x.

Committee Series Quality. Let us fix a *committee scoring function* $\mathrm{sc}\colon 2^C \to \mathbb{N}$, which assigns a nonnegative natural *committee score* to each *committee* $W \subseteq C$. Given an (ordered) *committee series* $\mathcal{W} = (W_1, W_2, \dots, W_\tau)$ of τ same-sized committees, $\mathrm{util}(\mathcal{W}) := \sum_{i \in [\tau]} \mathrm{sc}(W_i)$ is the *utilitarian committee series quality* of $\mathcal{W}$ and $\mathrm{egal}(\mathcal{W}) := \min_{i\in[\tau]} \mathrm{sc}(W_i)$ is its *egalitarian committee series quality*. We study *consecutive f-frequency committee series*. In such series, each candidate participates in at most f consecutive committees of a series.

Ordinal committee scoring functions. Let $E = (C, V)$ be an arbitrary ordinal election with m candidates, n voters, and $W = \{w_1, w_2, \dots, w_k\} \subseteq C$ be a committee. Following Bredereck *et al.* [2020] we consider three (families of) ordinal committee scoring functions (1.) *Chamberlin–Courant* (CC), (2.) *egalitarian Chamberlin-Courant* (eCC), and (3.) *weakly separable scoring functions (which we denote collectively by $\mathcal{F}_{WS}$)* formally defined as:

(1) $\mathrm{CC}(W) := \sum_{i\in[n]} \max_{w\in W} (m - \mathrm{pos}_i(w))$

(2) $\mathrm{eCC}(W) := \min_{i\in[n]} \max_{w\in W} (m - \mathrm{pos}_i(w))$

(3) a function Q is weakly separable, i.e., $Q \in \mathcal{F}_{\mathrm{WS}}$, if it can be associated with some function $\phi\colon [m] \to \mathbb{N}_0$ such that $Q(W) = \sum_{i\in[n]} \sum_{w\in W} \phi(\mathrm{pos}_i(w))$

The family of weakly-separable functions is very general. Among others, it includes such prominent voting rules as Plurality or Borda. Their respective ϕ functions are $\phi_{\text{plu}}(x) := \max(0, 2 - x)$ and $\phi_{\text{Bor}}(x) := m - x$ (both defined for $x \in [m]$).

Approval committee scoring functions. Similarly, for an arbitrary approval election $E = (C, V)$ with n voters and some committee $W \subseteq C$, we consider (1.) *(approval) Chamberlin-Courant* (AppCC), (2.) *threshold-α Chamberlin-Courant* (trCC^γ), for rational $\gamma \in (0, 1]$, and (3.) *approval score* (App), defined formally as follows:

(1) $\mathrm{AppCC}(W) := |\{v_i \in V : A(i) \cap W \neq \emptyset\}|$

(2) $\mathrm{trCC}^\gamma(W) := \begin{cases} 1 \text{ if } |\{v \in V \mid A(v) \cap W \neq \emptyset\}| \geq \gamma n, \\ 0 \text{ otherwise.} \end{cases}$

(3) $\mathrm{App}(W) := \sum_{v\in V} A(v) \cap W$

Central problem. We focus on the following computational problem, which we define very generally. We use $\alpha \in \{\mathrm{util}, \mathrm{egal}\}$ as a placeholder for a committee series quality measure and $\beta \in \{\mathrm{CC}, \mathrm{eCC}, \mathrm{AppCC}, \mathrm{trCC}^\gamma, \mathrm{App}\} \cup \mathcal{F}_{\mathrm{WS}}$, to indicate a committee scoring function. For readability, we directly substitute β with $\mathcal{F}_{\mathrm{WS}}$, when we mean that β is an arbitrary weakly-separable committee scoring function (e.g., egal-$\mathcal{F}_{\mathrm{WS}}$-SCE is a placeholder for any problem egal-g-SCE where $g \in \mathcal{F}_{\mathrm{WS}}$). We do not explicitly specify whether the input consists of ordinal or approval votes. It is to be inferred from the committee scoring function β in question.

α-β-SUCCESIVE COMMITTEES ELECTION (α-β-SCE)
Input: Election $E = (C, V)$ with candidates C and voters V, a number τ of committees in a target series, a size k

[1] Their taxonomy classifies successive committee elections as the setting of **O**rdered **O**ne profile **M**ultiple solutions (O-OM).

[2] For example: approvals are either complete rankings with ties or can be constructed from rankings by letting each voter approve a number of their top candidates.

of committees in a target series, a maximum candidate frequency f, and a minimal committee quality η.
Question: Is there a consecutive f-frequency committee series $\mathcal{S}$ of size τ consisting of size-k committees such that $\alpha(\beta(\mathcal{S})) \geq \eta$?

Bredereck *et al.* [2020] show that egal-β-SCE is NP-hard for all studied β even when simultaneously $k = 3$ and $f = 1$. Except for $\beta' \in \{\text{App}, \mathcal{F}_{\text{WS}}\}$, they prove the same for util-β'-SCE. Importantly, these two results immediately exclude efficient parameterized algorithms for small values of k, f, and their sum.

Even though the above formulation is a decision problem, all our algorithms can be used to find a requested committee series. In theorem statements, we give asymptotic running times of algorithms ignoring mostly irrelevant terms polynomial in the input. For clarity, we stress it using $\mathcal{O}^\star$ instead of the standard $\mathcal{O}$. The proofs marked by ★, or their parts, are deferred to the full version of our paper [Jain and Kaczmarczyk, 2025].

3 The Case of Few Candidates

Given the general computational hardness results of Bredereck *et al.* [2020], we start our search of efficient algorithms from cases with a bounded number of candidates. As argued in the introduction, this assumption can naturally be justified from the practical point of view. Parameter "number of candidates," which we denote by m, is too a standard parameter in the literature on the complexity of election problems.

For better accuracy, in some subsequent results (and in Table 1) we give running times using the size k of committees. Because $k \leq m$, such results always yield fixed-parameter tractability for parameter m. Naturally, they also show fixed-parameter tractability for parameter $m + k$. Recall that the hardness results of Bredereck *et al.* [2020] exclude fixed-parameter tractability for k alone.

Finding the winner of every multiwinner voting rule that assigns a score to each committee and chooses the one with the maximum score as the winner is fixed-parameter tractable with respect to the number of candidates (assuming computing the score of a committee is polynomial-time solvable). This observation becomes easy, when one realizes that for such rules it is enough to enumerate all possible committees and compute their scores (for example, see the works of Procaccia *et al.*; Betzler *et al.* [2008; 2013]) . However, a committee series is composed of multiple committees whose interdependency is nontrivially governed by the frequency of candidates and the requirement of consecutiveness. Indeed, even if we enumerate all at most 2^m committees, to construct a committee series we need to consider that each of them possibly come in any between one and frequency-many copies as demonstrated in Example 3 by Bredereck *et al.* [2020].

Does this complication to the domain of all possible solutions make finding the right solution computationally harder with respect to parameter "number of candidates"?

For the good news, we answer the above in negative. Quite surprisingly, even though the number of all possible committee series depends nonlinearly on the candidate frequency, the following series of results show fixed-parameter tractability of the problems we study with respect to solely the number of candidates. By this, the results reveal that the increase of the complexity of the domain of solutions is, intuitively, still bounded by a function of the number of candidates.

We start with a foundational result for the case of $f = 1$. While the result applies to both the utilitarian and egalitarian variants of our problem, it is the latter variant for which the result gives fixed-parameter tractability.

Theorem 1 (★). *There exists an algorithm that solves* util-β-SCE *and* egal-β-SCE *in* $\mathcal{O}^\star(2^m)$ *time for* $f = 1$ *and all studied* β.

The importance of Theorem 1 for the egal-β-SCE problem lies in combining it with the subsequent observation by Bredereck *et al.* [2020].

Proposition 1 (Lemma 1 by Bredereck *et al.* [2020])**.** egal-β-SCE *with* $f \geq 2$ *can be reduced to* egal-β-SCE *with* $f = 1$ *in linear time.*

The reduction in Proposition 1 does not increase the number of candidates. Hence, with Theorem 1, Proposition 1 yields a general result about the egalitarian version of the problem, which shows the sought tractability for small number of candidates.

Theorem 2. *There exists an algorithm that solves* egal-β-SCE *in* $\mathcal{O}^\star(2^m)$ *time for all studied* β.

Generalizing Theorem 1 in the utilitarian case does not directly lead to such an optimistic result as that for the egalitarian case. Instead, we obtain dynamic programming (DP) algorithms whose running time upper-bounds increase exponentially with the increase of parameter f. To increase readability, we first state our result for $f = 2$.

Theorem 3. *There is an* $\text{FPT}(m)$*-algorithm running in time* $\mathcal{O}^\star(4^{2m})$ *solving* util-β-SCE *for* $f = 2$ *and all studied* β.

Proof. Let us define a Boolean function $F(A, j, s, G)$, which returns true if among candidates A, there is a series achieving quality at least s and consisting of j committees, each of size 2 such that in the j-th committee candidates from $G \subseteq A$ appear for the first time in the series; F returns false otherwise. Our algorithm computes the values of F and returns true, if at least one of $F(C, \tau, \eta, G')$, for $G' \subseteq C$, $|G'| \leq k$ is true.

We compute the values of F applying the DP approach to the following recursive formula. Denoting by $\mathcal{W}(A, G, k)$ a collection of all size-k sets W such that $G \subseteq W \subseteq A$ we get:

$$f(A,j,s,G) = \bigvee_{W \in \mathcal{W}(A,G,k)} f(A \setminus G, j-1, s - \text{sc}(W), W \setminus G).$$

The correctness of the formula follows from the equation. Recall that to compute the left hand side value, we need to check whether it is possible to obtain exactly j 2-consecutive committees yielding quality s such that candidates from G appear in the j-th committee for the first time. To do so, in the formula, we consider each possible committee W containing G as a subset and scoring $\text{sc}(W)$. For each such W, we check whether there is a series of $j - 1$ consecutive committees of the requested size that jointly get quality $s - \text{sc}(W)$ consisting of candidates in $A \setminus G$. The last conditions follows from

	candidates number m	
egal-β-SCE	$\mathcal{O}^\star(2^m)$	Thm. 1
util-β-SCE	$\mathcal{O}^\star(m!(k-1)^m)$	Thm. 5
	$\mathcal{O}^\star(2^m)$; const. k and $f=1$	Thm. 1
	$\mathcal{O}^\star(4^{fm})$; any f	Thm. 4

	time horizon τ (for const. k)	
egal-β-SCE	$\mathcal{O}^\star(2.851^{(k-0.5501)\tau})$	Thm. 9
util-β-SCE	$\mathcal{O}^\star(2.851^{(k-0.5501)\tau})$; $f=1$	Thm. 6
	$\mathcal{O}^\star(2^{k\tau(f+1)}(2e)^{k\tau}(k\tau)^{\log(k\tau)})$; const. f	Thm. 7

Table 1: Our results for egalitarian and utilitarian committee series quality functions and $\beta \in \{\mathrm{CC}, \mathrm{eCC}, \mathrm{AppCC}, \mathrm{trCC}^\gamma, \mathrm{App}\}$. In all cases we assume $k \geq 3$. Note that Bredereck *et al.* [2020] show polynomial-time algorithms for util-App-SCE and egal-$\mathcal{F}_{\mathrm{WS}}$-SCE, so in these cases our algorithms run (much) slower.

the fact that candidates in G must be used for the first time only in W in question. Furthermore, we assure that the $j-1$-th committee contains the candidates that appear in W for the second time, as these have to take part in the committees consecutively; hence the last argument $W \setminus G$ of F's evaluation on the right-hand side. To conclude the proof, we define $F(A, 0, s, G) = \mathsf{true}$ for $s \leq 0$ and false for $s > 0$. We let the value of the function be false each time $\mathcal{W}(A, G, k) = \emptyset$.

There are at most $\tau 2^m \mathrm{poly}(n, m)$ values of F to compute, where the polynomial term comes form the maximum score of our committee scoring functions. It takes at most 2^m to compute a single value. Hence, as a result we obtain the claimed running time of $\mathcal{O}^\star(4^{2m})$. □

The above approach readily generalizes to arbitrary values of f by defining F to take arguments G_1 up to G_{f-1} instead of just G. Increasing the parameter space, leads to the exponential in f increase of the running time.

Theorem 4 (★). *There is an* $\mathrm{FPT}(m)$*-algorithm running in time* $\mathcal{O}^\star(4^{fm})$ *that solves* util-β-SCE *for all studied* β.

In the remainder of this section, we present Theorem 5 covering the claimed fixed-parameter tractability for m for util-β-SCE. There is a clear theoretical advantage of this next result over the one from Theorem 4, as the latter depends exponentially on both m and f. As we shall see, however, the running time of the algorithm from Theorem 5 may be (asymptotically) as large as m^m. Such a running time would (asymptotically) be way larger than 4^{fm} coming from Theorem 4 for small values of f. Hence, the algorithm from Theorem 4 (as well as the one from Theorem 1) might be more useful in practice. Especially since one of the motivations of studying candidate frequencies is to avoid excessive overload of committee members, which leads to small values of f [Bredereck *et al.*, 2020]. Finally, the algorithm from the following Theorem 5, albeit interesting from the computational complexity classification perspective, would rather turn out too computationally intensive to be applicable in practice.

Theorem 5. *There is an* $\mathrm{FPT}(m)$*-algorithm which runs in time* $\mathcal{O}^\star(m!(k+1)^m)$ *and solves* α-β-SCE *for* $\alpha \in \{\mathrm{util}, \mathrm{egal}\}$ *and all studied* β.

Proof. Throughout the whole proof we let k be the size of committees in a requested f-consecutive committee series of size τ. Our algorithm computes the sought committee series by repeatedly running a dynamic programming (DP) procedure over a collection of guesses, each of which represents a subspace of the solution space. The proof comes in three parts. We first discuss the guesses, then the DP procedure, and finally we show how to effectively enumerate the guesses.

Guesses. Our algorithm repeatedly guesses a specific structure over candidates C, which we call a *division*, and which describes a space of possible f-frequency committee series of size τ. Before we describe divisions, let us first consider some order of candidates represented by a bijection $\rho\colon [m] \to C$; e.g., $\rho(2) = c'$ means that candidate $c' \in C$ is ordered second. A subset X of candidates *is an interval* (according to ρ) if the set $\{i \colon \rho(i) \in X\}$ is an interval; intuitively, the candidates from X form an interval according to ρ. If X is an interval, then we denote by $\mathrm{beg}(X)$ the position in ρ of the leftmost candidate of X.

Definition 1. *A division $\mathcal{D}$ of size d is a* 3*-tuple* $(\rho, \mathcal{S}, \mathcal{F})$ *with* order ρ *of candidates, an ordered set* $\mathcal{S} = \{S_1, S_2, \dots S_d\}$ *of size-k committees called* primitives*, a collection* $\mathcal{F} = \{F_1, F_2, \dots, F_d\}$ *of* interim candidate sets *such that jointly:*

a) S_i *is an interval according to* ρ *for each* $i \in [d]$,
b) $\mathrm{beg}(S_i) < \mathrm{beg}(S_{i+1})$ *for each* $i \in [d-1]$,
c) $F_d = \emptyset$,
d) $F_i \cap F_j = \emptyset$ *for each* $(i, j) \in [d] \times [d]$, $i \neq j$,
e) $|F_i| \in \{0, k - |S_i \cap S_{i+1}|\}$ *for each* $i \in [d-1]$,
f) $S_i \cap S_{i+1} = \emptyset \Rightarrow F_i = \emptyset$ *for each* $i \in [d-1]$, *and*
g) $\left(\bigcup_{i\in[d]} F_i\right) \cap \left(\bigcup_{i\in[d]} S_i\right) = \emptyset$.

Together with the following intuitive description of the above definition, we provide an example division in Figure 1. Let us fix some order ρ. Then, the definition says that the primitives in $\mathcal{S}$ are d pairwise different interval size-k committees ordered increasingly by their interval beginning according to ρ. Further, the interim candidate sets are pairwise disjoint and consist of candidates that do not belong to any primitive committee. Interim candidate set F_d is always empty, and each interim candidate set F_i, $i \in [d-1]$ can be non-empty only if $Y = S_i \cap S_{i+1}$ is non-empty; but if F_i is non-empty, then it contains $k - |Y|$ elements.

A division is intended to encode a (sub)space of feasible solutions that can be used for an efficient search for a solution to our problem. So, it is crucial that at least one division describes a space containing the solution, if the latter exists.

Lemma 1 (★). *For each instance of* α-β-SCE*, there is a division* $\mathcal{D}$ *whose space contains a committee series maximizing* $\alpha(\beta(\mathcal{W}))$ *for the given values of* f*,* τ*, and* k*.*

The proof of the above lemma is constructive, but it does not offer an efficient way of finding an optimal solution. Be-

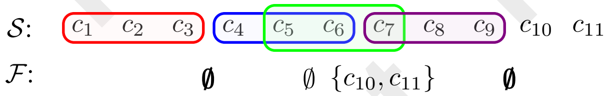

Figure 1: Division $\mathcal{D} = (\rho, \mathcal{S}, \mathcal{F})$ of size $d = 4$ of 11 candidates for a committee size $k = 3$ and order $\rho(i) = c_i$. The division features $\mathcal{S} = (\{c_1, c_2, c_3\}, \{c_4, c_5, c_6\}\{c_5, c_6, c_7\}, \{c_7, c_8, c_9\})$ and $\mathcal{F} = \{\boldsymbol{\emptyset}, \emptyset, \{c_{10}, c_{11}\}, \boldsymbol{\emptyset}\}$. Bolded $\boldsymbol{\emptyset}$ are empty by definition.

Figure 2: Two selected 3-consecutive 3-committee series of size $\tau = 6$ from the space of division $\mathcal{D}$ of Figure 1. Each column represents a single committee in a committee series, and each color represents a different primitive (or its part).

fore showing such a way, we present an intuition of how the elements of the space of some division $\mathcal{D}$ look like. To ease the presentation, we assume the identity order $\rho(i) = c_i$ for $i \in [m]$, and lay out some elements of the space in Figure 2.

First, observe that some consecutive f-frequency committee of size τ can be constructed as follows. We assign each primitive $S_i \in \mathcal{S}$ the respective number r_i of copies of this primitive ensuring that the sum of r_i's is exactly τ. Assuming that our r_i's do not violate the frequency limit f, we get a committee series $\mathcal{W}$ of size τ by repeating, in order, each primitive S_i exactly r_i times. Clearly, $\mathcal{W}$ is consecutive due to the interval requirements on the primitives in $\mathcal{S}$. There are, however, more committee series similar to $\mathcal{W}$ in the space of $\mathcal{D}$. They use the so-far ignored interim candidate sets $\mathcal{F}$. For an example, consider the following case. Assume that $\mathcal{W}$ contains a sequence S_i, S_i, S_{i+1} of neighboring committees and that $Y = S_i \cap S_{i+1}$ is non-empty. If $F_i \neq \emptyset$, then we can substitute the "middle" S_i, with a committee $S' = Y \cup F_i$, thus obtaining a new committee series $\mathcal{W}'$ containing a sequence S_i, S', S_{i+1}. Note that $\mathcal{W}'$ remains f-consecutive because by definition $|F_i| = k - |Y|$ and $F_i \cap Y = \emptyset$.

The whole space of $\mathcal{D}$ consists of the committee series emerging from all *valid* choices of the multiplicities r_i and from all valid substitutions (perhaps multiple at once) analogous to the construction of $\mathcal{W}'$.

Finding a Solution. Assume some division $\mathcal{D} = (\rho, \mathcal{S}, \mathcal{F})$, where $\mathcal{S} = (S_1, S_2, \ldots, S_d)$ and $\mathcal{F} = (F_1, F_2, \ldots, F_d)$. We show a dynamic programming algorithm (DP) running in the claimed FPT-time that tests whether the space of $\mathcal{D}$ contains a sought committee. We define a Boolean function $T(t, i, q, u, j, r, p)$, for $q > 0$, $r > 0$, which is true if and only if there is an f-consecutive series X such that:

a) X contains t committees (of size k),
b) primitive $S_i \in \mathcal{S}$ is repeated exactly q times in X,
c) if $F_i \neq \emptyset$, committee $S' = F_i \cup (S_i \cap S_{i+1})$, called *$i$-intermediate*, is repeated u times in X between the copies of S_i and S_{i+1},
d) in which candidate $\rho(j) \in S_i$ is part of exactly r consecutive committees of X, and
e) $\alpha(\beta(X)) \geq p$.

Assuming that $\rho(j')$ is the last candidate in S_d according to ρ, there exists a sough f-consecutive series of size τ achieving a committee series quality η if there is a pair $(q', r') \in [f] \times [f]$ for which $T(\tau, d, q', 0, j', r', \eta)$ outputs true.

We recursively compute the values $T(t, i, q, u, j, r, p)$ in the order of increasing values of t, i, j, and p; the variables are mentioned in the order of increasing frequency of change during computation iterations. Because the candidates are ordered with respect to ρ, it is convenient to think of combinations of values i and j as cells in a grid, where j is the column number representing candidate $\rho(j)$ and i is the row number representing primitive $S_i \in \mathcal{S}$. In the grid, primitive S_i forms an interval in the i-th row; this interval corresponds to the candidates in S_i. As depicted in Figure 3, by convention, we place point $(1, 1)$, corresponding to candidate $\rho(1)$ in primitive S_1, in the upper left corner of the corresponding grid.

Some values of $T(t, i, q, u, j, r, \eta)$ are invalid by definition or in an obvious manner. This is the case for:

(I1) each i and $j \in [m]$ for which $\rho(j) \notin C_i$; see dark shaded cells in Figure 3,
(I2) cases in which $q + t > \tau$, $q + u > f$, or $r > f$,
(I3) positive values u for i for which $F_i = \emptyset$; in particular, by definition of $\mathcal{D}$, this holds when $S_i \cap S_{i+1} \neq \emptyset$.

For readability, we never explicitly test for these invalid cases assuming, for technical reasons, that T returns false for them.

For the sake of presentation, for each primitive $S_i \in \mathcal{S}$, we define the *head* and the *tail* of S_i as, respectively, the first and the last candidate of S_i with respect to order ρ (as demonstrated in Figure 3). Formally, $\mathrm{head}(S_i) = j'$ and $\mathrm{tail}(S_i) = j''$ if for each $j \in [m]$ such that $\rho(j) \in S_i$, $j' \leq j$ and $j \leq j''$. If a candidate is neither the tail nor the head of some primitive $S_i \in \mathcal{S}$, then it is *intermediate* in this primitive. Note that a candidate might be the head of one primitive but an intermediate candidate for another primitive.

In what follows, we frequently consider a situation in which we seek a committee series of quality at least p by repeating a primitive $S_i \in \mathcal{S}$ exactly q times and the committee $(S_i \cap S_{i+1}) \cup F_i$ exactly u times. In such cases, we use $\mathrm{rq}(i, p, q, u) \coloneqq p - q\beta(S_i) - u\beta((S_i \cap S_{i+1}) \cup F_i)$ that describes the required quality of a to-be-constructed committee series prior to using the $q + u$ committees mentioned above. This is where the subtraction in the expression comes from. Our formulas use $u = 0$ whenever $(S_i \cap S_{i+1}) \cup F_i$ is not a committee of the requested size (when either $F_i = \emptyset$ or $S_i \cap S_{i+1} = \emptyset$). Consequently, the related term vanishes.

We continue with presenting a recursive formula for T, which we split into multiple cases for readability. Values of the base cases $T(t, 1, q, u, j, r, p)$ follow directly from the definition of T.

We first focus on such cases $T(t, i, q, u, j, r, p)$ in which $\mathrm{head}(S_i) = j$ and $S_{i-1} \cap S_i = \emptyset$. We skip computation at all and return false if $q \neq r$. Indeed, $q = r$ is required as all candidates in S_i will appear for the first time in the constructed committee series.

$$T(t, i, q, u, j, r, p) = \bigvee_{q' \in [f], r' \in [f]} T(t - q - u, i - 1, q', 0, \mathrm{tail}(S_{i-1}), r', \mathrm{rq}(i, p, q, u))$$

In words, the formula's alternative tests whether we can achieve sufficient committee quality $\mathrm{rq}(i, p, q, u)$, *before* we repeat primitive S_i exactly q times and committee $S_i \cap S_{i+1} \cup F_i$ exactly u times. Since $S_{i-1} \cap S_i = \emptyset$ implies that $F_{i-1} = \emptyset$, in the alternative we invoke T with the respective parameter equal to 0. Recall that expression $\mathrm{rq}(\cdot)$ remains correct even if $F_i = \emptyset$, which by Case (I3) implies $u = 0$.

Next, we consider $\mathrm{head}(S_i) = j$ under assumption that $S_{i-1} \cap S_i \neq \emptyset$. Consequently, $q < r$, as the j-th candidate belongs to $S_{i-1} \cap S_i$, and S_{i-1} has to be repeated at least once (see the domain of T). So, if $q \geq r$ we return false and otherwise we compute T as follows:

$$T(t, i, q, u, j, r, p) = \bigvee_{q' \in [f], u' \in [f]} T(t - q - u, i - 1, q', u', j, r - q, \mathrm{rq}(i, p, q, u)).$$

This time, we ensure that before using primitive S_i and interim candidates F_i, we used the j-th candidate exactly $r - q$ times. We do not subtract u, as by the fact that primitives are different, j-th candidate cannot be part of $(S_i \cap S_{i+1}) \cup F_i$, which is repeated u times. Accordingly, the alternative iterates over all possible values of q' and u' to check if there is any committee series using primitives up to S_{i-1} of the required quality (see the discussion in the previous case).

We continue with computing $T(t, i, q, u, j, r, p)$ for cases, in which the j-th candidate is either an intermediate or a tail candidate. Consequently, the j-th candidate (from the respective primitive $S_i \in \mathcal{S}$) can be part of ℓ-intermediate committees for $\ell \geq i$, i.e., that come after all copies of S_i in the built committee series $\mathcal{W}$. Previously, when the j-th candidate was a head candidate, it could only be part of the ℓ-intermediate committees for $\ell < i$. For this reason, the following formulas sometimes impose additional conditions.

We start with a simple case. Let the j-th candidate $c = \rho(j)$ be the intermediate or tail of primitive S_i. Assume that $c \notin (S_i \cap S_{i+1})$.

$$T(t, i, q, u, j, r, p) = \bigvee_{r' \in \{r, r+1, \ldots, f\}} T(t, i, q, u, j - 1, r', p)$$

The number repetitions of S_i and the score p is always fixed during the computation for S_i's head candidate. So in this case the alternative recursively mostly "carries on" function values from previous candidates of S_i. We iterate using r' because the $j - 1$-th candidate could be part of more (previous) committees than candidate c. Due to our interval requirements and sorting of $S_i \in \mathcal{S}$, we know that $r' \geq r$.

Next, consider computing $T(t, i, q, u, j, r, p)$ assuming that the j-th candidate $c = \rho(j)$ is an intermediate or tail candidate, $c \in S_i \cap S_{i+1}$, and $c \notin S_{i-1}$. We repeat c exactly $q + u$ times as a member of q copies of S_i and of u copies of i-intermediate committee. Hence, we immediately return false if $r \neq q + u$, and otherwise we have

$$T(t, i, q, u, j, r, p) = \bigvee_{r' \in \{q, q+1, \ldots, f\}} T(t, i, q, u, j - 1, r', p).$$

The lower-bound of iterator r' follows from the fact that the $j - 1$-th candidate must be repeated at least as many times as the primitive S_i to which the candidate belongs is repeated.

In the final case we consider the, intermediate or tail, j-th candidate $c = \rho(j)$ such that $c \in S_i \cap S_{i+1}$, and $c \in S_{i-1}$.

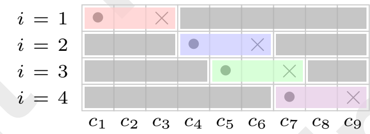

Figure 3: A grid-representation of division $\mathcal{D}$ from Figure 1 with the heads (●) and tails (×) of the primitives. Shaded boxes represent invalid value combinations of parameters i and j of function T. For readability, we omit the sets of interim candidates.

Since c is used in copies of primitive S_{i-1}, we return false if $q + u \leq r$. Otherwise, we compute T as follows:

$$T(t, i, q, u, j, r, p) = \bigvee_{r' \in \{q, q+1, \ldots, f\}} T(t, i, q, u, j - 1, r', p) \wedge \bigvee_{q' \in [f], u' \in [f]} T(t - q - u, i - 1, q', u', j, r - q - u, \mathrm{rq}(i, p, q, u)).$$

On top on all conditions of the counterpart case with $c \notin S_{i-1}$ described directly above this case, we add further requirements. Observe that here c is used $q + u$ times but it was also previously used. The second alternative of the formula ensures that these $q + u$ uses of candidate c would not make c exceed the frequency f.

The case distinctions is exhaustive so the correctness of the formula stems directly from its description. To compute each value of function T, we need at most $O(f^2)$ steps. Further, for some division $\mathcal{D}$ of size d, we need to compute $d \cdot k\tau f^3 \cdot \mathrm{poly}(n, m)$ values of T, as there is dk valid pairs of parameters i and j, $t \in [\tau]$, $(q, u, r) \in [f]^3$. Hence, we obtain a polynomial running-time for each division.

Constructing Guesses and Running Time. We run the DP procedure for each possible division $\mathcal{D} = (\rho, \mathcal{S}, \mathcal{F})$. We first select some order ρ of candidates out of $m!$ of them. Then, we guess the division size d. Note that $\mathcal{S} = (S_1, S_2, \ldots, S_d)$ can be represented as a vector $(s_1, s_2, \ldots, s_d)$, where s_i, $i \in [d]$, denotes the rightmost candidate of the interval of committee S_i given ordering ρ. Hence, the guessed division size d varies from one to at most m, where the upper bound is not tight and describes the maximum number of distinct equisized interval committees coming from taking $d = m - k$ and $s_i = k + i$, for $i \in [d]$. Importantly, τ does not contribute to the number of possible values of d, as it is only the DP procedure that achieves τ committees for some $\mathcal{D}$. Next, we guess each entry of the $\mathcal{S}$ vector, except for the first one which is fixed. To this end, we choose the value of each s_i, $i \in \{2, 3, \ldots, d\}$, from range $\{s_{i-1} + 1, s_{i-1} + 2, \ldots, s_{i-1} + k + 1\}$ (recall that all primitives in $\mathcal{S}$ have to form an interval according ρ). Finally, for each non-empty intersection X of neighboring primitives, of which there are at most d, we either select an empty interim set or take next unused $k - |X|$ candidates according to ρ (as long as the latter is possible). This totals in at most $m! \cdot m \cdot (k+1)^m \cdot 2^m$ guesses. Overall, the algorithm is clearly fixed-parameter tractable with respect to m. □

The approach behind Theorem 5 in fact works for any quality measures of committee series quality and committee scoring function as long as their values are polynomially bounded by the input size. This leads to a general result covering much more scoring functions than those described in Section 2.

Corollary 1. *There is an* $\mathrm{FPT}(m)$*-algorithm with running time* $\mathcal{O}^\star(m!(k-1)^m)$ *solving* α-β-SCE *for arbitrary computable aggregation* α *and a committee scoring function* β *whose values are polynomially bounded in the instance size.*

4 The Case of Short Time Horizon

The algorithms from the previous section are no longer of use for large pools of candidates. In such cases, approaches tailored to small values of parameter τ, which is a formal framing of short time horizon in our model, might come to the rescue. Imagine, for example, an online streaming platform building daily music recommendation for the following week, based on a user's past-week activity. While such a task might likely include even hundreds of songs, it is still only seven lists (committees) that we want to arrange in a series.

In most cases that we study, however, our expectations cannot be stretched as far as obtaining FPT-algorithms for parameter τ. The Chamberlin–Courant rule, in its variants for approval as well as ordinal preferences, is W[2]-hard for parameter "committee size" already in the multiwinner voting model [Lu and Boutilier, 2011; Betzler *et al.*, 2013]. Because these results are special cases of α-β-SCE for both committee series quality variants and $\beta \in \{\mathrm{CC}, \mathrm{trCC}^\gamma, \mathrm{eCC}, \mathrm{AppCC}\}$ for $\tau = 1$, there is no hope for fixed-parameter tractability for parameter $k + \tau$ in this case.

Consequently, we focus on FPT-algorithms with respect to τ assuming a constant committee size. On the one hand, such results are a clear advancement from the theoretical perspective given that util-β-SCE and egal-β-SCE both are NP-hard even for the constant committee size for almost every committee scoring β that we study [Bredereck *et al.*, 2020] (the exceptions are util-$\mathcal{F}_{\mathrm{WS}}$-SCE and util-App-SCE). On the other hand, constant values of k are still interesting from the practical point of view as small committee sizes might appear in practice. For example, in our toy example about daily music recommendation, providing too many songs each day would be overwhelming for the user and ineffective in promoting featured tracks.

Our first algorithm reinterprets util-β-SCE with $f = 1$ as an instance of WEIGHTED SET PACKING and then applies a known procedure of Goyal *et al.* [2015] for solving the latter.

Theorem 6 (★). *There exists an algorithm that solves* util-β-SCE *in* $\mathcal{O}^\star(2.851^{(k-0.5501)\tau})$ *time for all studied* β *when the committee size* k *is constant and* $f = 1$.

Next, we prove the result for constant $f \geq 2$. Without loss of generality, we assume that $\tau > f$. Otherwise, we find a committee of maximum score in polynomial time and repeat it f times. Since k is a constant, computing a maximum score committee is polynomial-time solvable for every committee scoring β that we study.

Theorem 7 (★). *There exists an algorithm that solves* util-β-SCE *in* $\mathcal{O}^\star\left(2^{k\tau(f+1)}(2e)^{k\tau}(k\tau)^{\log(k\tau)}\right)$ *time for all studied* β *when the committee size* k *and* f *are constant.*

The proof of Theorem 7 heavily depends on the following randomized algorithm.

Theorem 8 (★). *There exists a randomized algorithm that given an instance of* util-β-SCE *for all studied* β *either reports* failure *or outputs a solution in* $\mathcal{O}^\star(2^{k\tau(f+1)}(2e)^{k\tau})$ *time. Moreover, if the algorithm is given a* yes*-instance, it returns* yes *with probability at least* $1/2$, *and if the algorithm is given a* no*-instance, it returns* no *with probability 1.*

Proof sketch. A solution $\mathcal{S} = (S_1, \ldots, S_\tau)$ is called *colorful* if every pair of candidates s, s' in $\biguplus_{i\in[\tau]} S_i$ has distinct colors. Now, our algorithm runs in three phases. In the first phase, we uniformly and independently at random color the candidates forming our solution. Then, we discard the sets that cannot be part of our solution under the coloring. In the third phase, we use dynamic programming to turn the coloring into a colorful solution, if the coloring admits one. □

We obtain Theorem 7 by derandomizing the algorithm of Theorem 8. To this end we employ a (p, q)*-perfect hash family* [Alon *et al.*, 1995] that we construct in the requested time applying the results of Naor *et al.* [1995] and Cygan *et al.* [2015].

We provide more good news, by presenting an analogous result for the egalitarian committee series evaluation. By this, we cover all of our cases of interest (for constant k).

Theorem 9 (★). *There exists an algorithm that solves* egal-β-SCE *in* $\mathcal{O}^\star(2.851^{(k-0.5501)\tau})$ *time for all studied* β *when the committee size* k *is constant.*

5 Conclusions and Future Directions

We have provided first algorithms for solving hard instances emerging from the model of successive committee elections [Bredereck *et al.*, 2020]. Extending the algorithmic understanding of a recent area of temporal elections, our results concern potentially practical scenarios including small numbers of candidates and a short time horizon. In the light of increasing resonance of experiments in computational social choice [Boehmer *et al.*, 2024], our algorithms potentially enable experimental study on successive committee elections.

Theoretical follow-up directions include complementing our picture with parameterizations by "number n of voters" and "committee series quality η." While both these parameters might turn out to be quite large in practical applications, obtaining the respective results would complete the picture of the parameterized complexity of the studied problems. We note that parameter η in part inherits technical intricacies of the study of the parametrization by (mis)representation for Chamberlin–Courant rules [Betzler *et al.*, 2013; Chen and Roy, 2022]. Importantly, Bredereck *et al.* [2020, Theorem 5] implicitly excludes fixed-parameter tractability of η for egal-$\mathrm{trCC}^1(W)$-SCE, while our results yield it for util-CC-SCE and util-AppCC-SCE for η.

On the more practical side, our algorithms require an empirical treatment to verify its potential applicability in practice. As our study only gives pessimistic running-time upper bounds, a thorough investigation based on synthetic and real-life instances is needed to check the algorithms behavior on more realistic input instances.

Acknowledgements

We thank anonymous reviewers for their helpful comments. Pallavi Jain acknowledges her support from SERB-SUPRA (grant number SPR/2021/000860) and IITJ Seed Grant (grant no I/SEED/PJ/20210119). Andrzej Kaczmarczyk acknowledges support from NSF CCF-2303372 and ONR N00014-23-1-2802. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101002854).

References

[Alon *et al.*, 1995] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM (JACM)*, 42(4):844–856, 1995.

[Betzler *et al.*, 2013] Nadja Betzler, Arkadii Slinko, and Johannes Uhlmann. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013.

[Boehmer and Niedermeier, 2021] Niclas Boehmer and Rolf Niedermeier. Broadening the research agenda for computational social choice: Multiple preference profiles and multiple solutions. In *Proceedings of AAMAS-2021*, pages 1–5, 2021.

[Boehmer *et al.*, 2024] Niclas Boehmer, Piotr Faliszewski, Lukasz Janeczko, Andrzej Kaczmarczyk, Grzegorz Lisowski, Grzegorz Pierczynski, Simon Rey, Dariusz Stolicki, Stanislaw Szufa, and Tomasz Was. Guide to numerical experiments on elections in computational social choice. In *Proceedings of IJCAI-2024 (Survey Track)*, pages 7962–7970, 2024.

[Bredereck *et al.*, 2020] Robert Bredereck, Andrzej Kaczmarczyk, and Rolf Niedermeier. Electing successive committees: Complexity and algorithms. In *Proceedings of AAAI-2020*, pages 1846–1853, 2020.

[Bredereck *et al.*, 2022] Robert Bredereck, Till Fluschnik, and Andrzej Kaczmarczyk. When votes change and committees should (not). In *Proceedings of IJCAI-2022*, pages 144–150, 2022.

[Bulteau *et al.*, 2021] Laurent Bulteau, Noam Hazon, Rutvik Page, Ariel Rosenfeld, and Nimrod Talmon. Justified representation for perpetual voting. *IEEE Access*, 9:96598–96612, 2021.

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

[Chandak *et al.*, 2024] Nikhil Chandak, Shashwat Goel, and Dominik Peters. Proportional aggregation of preferences for sequential decision making. In *Proceedings of AAAI-2024*, pages 9573–9581, 2024.

[Chen and Roy, 2022] Jiehua Chen and Sanjukta Roy. Parameterized intractability for multi-winner election under the Chamberlin-Courant rule and the Monroe rule. arXiv:2202.12006 [cs.MA], 2022.

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[Deltl *et al.*, 2023] Eva Michelle Deltl, Till Fluschnik, and Robert Bredereck. Algorithmics of egalitarian versus equitable sequences of committees. In *Proceedings of IJCAI-2023*, pages 2651–2658, 2023.

[Elkind *et al.*, 2024a] Edith Elkind, Svetlana Obraztsova, and Nicholas Teh. Temporal fairness in multiwinner voting. In *Proceedings of AAAI-2024*, pages 22633–22640, 2024.

[Elkind *et al.*, 2024b] Edith Elkind, Svetlana Obraztsova, and Nicholas Teh. Verifying proportionality in temporal voting. In *Proceedings of AAMAS-2024*, pages 2246–2248, 2024.

[Faliszewski *et al.*, 2017] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Multiwinner voting: A new challenge for social choice theory. In U. Endriss, editor, *Trends in Computational Social Choice*. AI Access Foundation, 2017.

[Goyal *et al.*, 2015] Prachi Goyal, Neeldhara Misra, Fahad Panolan, and Meirav Zehavi. Deterministic algorithms for matching and packing problems based on representative sets. *SIAM Journal on Discrete Mathematics*, 29(4):1815–1836, 2015.

[Jain and Kaczmarczyk, 2025] Pallavi Jain and Andrzej Kaczmarczyk. Efficient algorithms for electing successive committees. arXiv:2505.18287 [cs.GT], 2025.

[Kellerhals *et al.*, 2021] Leon Kellerhals, Malte Renken, and Philipp Zschoche. Parameterized algorithms for diverse multistage problems. In *Proceedings of ESA-2021*, pages 55:1–55:17, 2021.

[Lackner and Maly, 2021] Martin Lackner and Jan Maly. Perpetual voting: The axiomatic lens. arXiv:2104.15058 [cs.GT], 2021.

[Lackner and Skowron, 2023] Martin Lackner and Piotr Skowron. *Multi-Winner Voting with Approval Preferences*. Springer, 2023.

[Lackner, 2020] Martin Lackner. Perpetual voting: Fairness in long-term decision making. In *Proceedings of AAAI-2020*, pages 2103–2110, 2020.

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

[Naor *et al.*, 1995] Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *Proceedings of FOCS-1995*, pages 182–191, 1995.

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