

Multiwinner Voting with Possibly Unavailable Candidates

Markus Brill^{1,2}, Hayrullah Dindar¹, Jonas Israel¹, Jérôme Lang³,
Jannik Peters¹, Ulrike Schmidt-Kraepelin¹

¹ Research Group Efficient Algorithms, Technische Universität Berlin, Germany

² Department of Computer Science, University of Warwick, Coventry, UK

³ CNRS, Université Paris-Dauphine, PSL, LAMSADE, France

markus.brill@warwick.ac.uk, hayrullah.dindar@tu-berlin.de, j.israel@tu-berlin.de, lang@lamsade.dauphine.fr,
jannik.peters@tu-berlin.de, u.schmidt-kraepelin@tu-berlin.de

Abstract

Selecting a committee that meets diversity and proportionality criteria is a challenging endeavor that has been studied extensively in recent years. This task becomes even more challenging when some of the selected candidates decline the invitation to join the committee. Since the unavailability of one candidate may impact the rest of the selection, inviting all candidates at the same time may lead to a suboptimal committee. Instead, invitations should be sequential and conditional on which candidates invited so far accepted the invitation: the solution to the committee selection problem is a *query policy*. If invitation queries are binding, they should be *safe*: one should not query a candidate without being sure that whatever the set of available candidates possible at that stage, her inclusion will not jeopardize committee optimality. Assuming approval-based inputs, we characterize the set of rules for which a safe query exists at every stage. In order to parallelize the invitation process, we investigate the computation of safe parallel queries, and show that it is often hard. We also study the existence of safe parallel queries with respect to proportionality axioms such as extended justified representation.

1 Introduction

In multiwinner voting the goal is to select a set of candidates based on preferences expressed by voters. In the usual setting there is a given set of candidates out of which a fixed-size subset has to be chosen. This, however, does not capture many real-life scenarios.

For instance, consider a university department hiring multiple researchers across multiple groups simultaneously. The hiring committee will view the researchers' applications, and based on the preferences of the members of the hiring committee, the university selects a subset of candidates to offer a position. However, of these suitable candidates, not all might accept the offer, after which new candidates need to be invited. When choosing these new candidates, it might play a prominent role which candidates initially accepted their offer and which declined. As an example, assume there are three groups in the computer science department: algorithms (ALG), machine learning (ML), and cryptography (CG), representing respectively $\frac{3}{8}$, $\frac{3}{8}$, and $\frac{1}{4}$ of the department; and four candidates c_1, c_2, c_3, c_4 . Candidate c_1 is interesting for

ALG and ML, candidate c_2 for CG, candidate c_3 for ALG, and candidate c_4 for ML. If the goal of the selection process is to maximize some notion of representation among the different groups, it is plausible that the preference over subsets of candidates is such that $\{c_1, c_2\}$ is best, and that $\{c_3, c_4\}$ is better than $\{c_2, c_3\}$ and $\{c_2, c_4\}$. Thus, it is optimal in the first stage to invite c_1 , and then to invite c_2 if c_1 is available and otherwise to invite c_3 . We see that whether a candidate is available or not influences the selection of further candidates.

There are two interpretations of availability queries. We will assume here that they are *binding*: once a candidate is queried, she is irrevocably included in the selection if she is available — think of inviting plenary speakers at a conference; it is not possible to revoke already accepted invitations. The case of *non-binding* queries is discussed in Section 5.

The context we described is a multiwinner variant of the *unavailable candidate model* (Lu and Boutilier 2010): voters express their preferences over a set of potential candidates, but only a subset of them will eventually be available; the output then is a ranked list of candidates, according to which candidates are queried; and the first one who is available is selected. Our context is similar but the need to select *several* candidates raises two complications: the query policy should be conditional on offers already accepted or rejected, and when selecting candidates to query, one should keep in mind proportionality targets bearing on the global selection.

Our contribution. In this paper, we formalize the mentioned model and adapt the concept of a *query policy* (Boutilier et al. 2014) to multiwinner elections. Such a policy queries candidates for their availability, with the constraint that a queried candidate has to be included in the selection once they signal their availability.

We then study common *approval-based committee (ABC)* rules and investigate whether these rules admit a *safe query policy*, i.e., a policy that only queries candidates that are guaranteed to be included in an optimal selection whenever they are available, whatever the other available candidates are. We show that under a mild technical assumption a necessary and sufficient condition for a rule to admit a safe query policy is “sequentiality” (to be formally defined later). This allows us to show, in particular, that Phragmén’s sequential rule and the Method of Equal Shares admit a safe query policy, and that Proportional Approval Voting (PAV) and other non-

sequential Thiele rules do not.

Further, we study the complexity of finding safe (sets of) candidates, i.e., candidates and sets of candidates that are always included in an optimal selection if they are available. Whereas this problem is trivial for standard Multiwinner Approval Voting (AV), we show that

- for all non-sequential Thiele rules other than AV, deciding whether a safe candidate exists is coNP-hard;
- for all sequential Thiele rules other than AV (for which a safe candidate always exists), deciding whether there exists a safe set containing more than one candidate is coNP-hard; and
- for all (sequential and non-sequential) Thiele rules other than AV, deciding whether a *given* candidate is safe is coNP-hard.

Finally, we turn to notions of proportionality, namely *proportional justified representation* (PJR) and *extended justified representation* (EJR). We show that for PJR, a set of k safe candidates always exists, and such a set is selected by the leximax-Phragmén rule. For EJR, we show that known rules satisfying EJR do not always select k safe candidates for EJR; however, we provide a query policy that finds a safe set of size at least two. Omitted proofs can be found in the full version of this paper (available at <https://www.markus-brill.de/>).

Related work. There is a large literature on uncertainty and communication in voting, see the survey by Boutilier and Rosenschein (2016). Uncertain knowledge can bear on agents’ preferences, on the voting rule, or on the set of running candidates. Our work deals with the latter type.

The unavailable candidate model was introduced independently by Lu and Boutilier (2010) and (with a different motivation) by Baldiga and Green (2013). In this model, only one candidate has to be selected; voters have ordinal preferences over potential candidates that are associated with probabilities of availability; and the aim is to build a ranked list such that the first available candidate in the list maximizes the probability of being the plurality winner. There have been a number of follow-ups. Oren, Filmus, and Boutilier (2013) assume that voters report top- k rankings instead of complete rankings. Boutilier et al. (2014) consider conditional query policies whose goal is to identify the winner with respect to the actual set of running candidates; these queries are non-binding and have costs. While Lu and Boutilier (2010) assume voters’ utilities are binary (1 if the best available candidate is selected, 0 otherwise), nonbinary utilities are considered by Grivet Sébert et al. (2021). See the full version of this paper for a structured comparison of these works, and Section 5 for a discussion on how to further extend our work to account for other specificities.

The dynamic selection of a set of candidates, where at each step one must decide whether to select a candidate irrevocably or not given approval votes, is studied by Do et al. (2022). A major difference with the works mentioned above is that the uncertainty is not related to candidates’ availability, but to the order in which they appear.

Fair sortition for the selection of a representative citizen assembly (Flanigan et al. 2021) also involves availability

queries, but they are non-binding and made offline, and then a (randomized) selection algorithm is called. Also, proportionality relates to demographic features, not to votes.

Our work is heavily related to ABC voting (Lackner and Skowron 2022). Uncertainty in ABC voting, albeit only for the case of uncertain preferences, has been discussed by Barrot et al. (2013), Terzopoulou, Karpov, and Obraztsova (2021), and Imber et al. (2022).

2 Setting

We first recall the standard *approval-based committee* (ABC) voting setting, where $N = [n]$ is a set of voters, $C = \{c_1, \dots, c_m\}$ is a set of candidates, $A = (A_1, \dots, A_n)$ is an *approval profile* containing the approval set $A_i \subseteq C$ for each voter $i \in N$, and $k \in [m]$ is the committee size.¹ For $c \in C$ we let $N_c = \{i \in N : c \in A_i\}$ be the *approvers* of c . A *committee* is a subset $W \subseteq C$ of candidates of size $|W| = k$. Since the approval profile implicitly gives the sets of voters and candidates, we omit N and C from the notation and refer to an ABC instance by the pair (A, k) .

Unavailable candidates. In our setting, some candidates in C might be unavailable. We assume there exists an underlying *set of available candidates* $X \subseteq C$. All candidates $c \in C \setminus X$ are *unavailable*. Instances and committees are defined as in the standard setting, except that we allow rules to output fewer than k candidates in case $|X| < k$.² Thus, in this more complex setting, a *committee* is a subset $W \subseteq C$ of candidates with $|W| = \min\{k, |X|\}$.

An ABC voting rule under uncertain availability is a function r that takes as input the approval profile (over all candidates C), the target committee size k , and the set of available candidates X and outputs one (or more) committees $W \in r(A, k, X)$ of size $\min\{k, |X|\}$. Every standard ABC rule can be translated to this setting in a straightforward way, by defining $r(A, k, X)$ as the committee selected by r when applied to $(A|_X, \min\{k, |X|\})$. We are interested in how we can obtain committees of only available candidates without direct access to the set of available candidates X . For that, we introduce the notion of *safe* (sets of) candidates.

Definition 1. *Given an ABC instance (A, k) and a voting rule r , a subset of candidates $C' \subseteq C$ is safe if for any set of available candidates X there is some committee $W \in r(A, k, X)$ such that $C' \cap X \subseteq W$. We say that candidate $c \in C$ is safe if $\{c\}$ is safe.*

That is, we consider C' safe if independent of which candidates are available, we can complete the available candidates among C' to an output of r . Note that for a set C' to be safe all (available) candidates in C' must be safe when taken together. This is stronger than simply enforcing that C' consists

¹For $t \in [m]$, we let $[t]$ denote the set $\{1, \dots, t\}$.

²In this case, we assume that the rule will return all available candidates. ABC rules with a variable number of winners have been considered by Freeman, Kahng, and Pennock (2020); a crucial difference with our setting is that they do not constrain the number of winners at all, while we try to choose a committee whose size is as close as possible to the target size k .

of candidates that are each safe on their own.³ In order to be able to talk about the safeness of policies, we extend the definition of safeness to also incorporate situations in which some candidates have already been selected. For this, we say that a subset $C' \subseteq C$ of candidates is *safe with respect to a sub-committee* $W' \subseteq C$ and a set of candidates known to be unavailable $U \subseteq C$ if, for any set of available candidates $X \supseteq W'$ with $X \cap U = \emptyset$, there is some committee $W \in \mathcal{R}(A, k, X)$ with $W \supseteq W'$. That is, we can always extend W' with the available candidates from C' to a winning committee. Similar to safeness for voting rules, we can also say that a set of candidates is safe for proportionality axioms like EJR and PJR (cf. Section 4.2) if we can extend the set to a committee that satisfies the axiom.

Query policies. A query policy is an algorithm that takes an approval profile and a target committee size k as input and produces a committee by iteratively querying the availability of candidates. In order to keep track of candidates whose availability has already been queried, we use Q^+ and Q^- to denote the set of *known available* candidates and *known unavailable* candidates, respectively. In particular, after each round t of the query policy, we have an *information set* $Q_t = (Q_t^+, Q_t^-)$, where $Q_t^+ \subseteq X$ is the set of available candidates that have already been queried and $Q_t^- \subseteq C \setminus X$ is the set of unavailable candidates that have been queried. By definition, we have $Q_0^+ = Q_0^- = \emptyset$ and $Q_t^+ \cap Q_t^- = \emptyset$ for all $t \geq 0$.

In each round $t \geq 1$, we let $c_t \in C$ denote the candidate that is queried for their availability by the policy. If c_t is available, we let $Q_t = (Q_{t-1}^+ \cup \{c_t\}, Q_{t-1}^-)$; otherwise, $Q_t = (Q_{t-1}^+, Q_{t-1}^- \cup \{c_t\})$. Without loss of generality, we can assume that every candidate is queried at most once. Thus $|Q_t^+ \cup Q_t^-| = t$ and the number of rounds is upper-bounded by $m = |C|$. A policy can terminate only if $|Q_t^+| \geq k$ or $|Q_t^+ \cup Q_t^-| = |C|$ (and, therefore, $Q_t^+ = X$). At termination, it outputs a committee $W \subseteq Q_t^+$ of size $|W| = \min\{k, |X|\}$.

We can think of a query policy as a labeled binary tree. Given an approval profile and committee size k , we construct a rooted binary tree of depth m corresponding to a query policy as follows. We label the root node of the tree with the candidate c_1 that is queried first by the policy. The left child of c_1 is labeled with the candidate that is queried by the policy if c_1 is available, and the right child with the candidate that is queried if c_1 is not available. We continue this labeling until the query policy terminates. A terminal node is labeled with the committee selected by the policy.

Example 1. Consider the approval profile where 3 voters approve $\{a, c\}$, 2 voters approve $\{a, d\}$, and one voter approves $\{b\}$. Say that a candidate c_j “covers” a voter i if $c_j \in A_i$ and consider the following query policy: At each round, query a candidate that covers the most uncovered voters, breaking ties in favor of candidates covering more already covered voters, and terminating when the target committee size is reached (or there are no more candidates to query). For the given profile and $k = 2$, the tree corresponding to this query policy is shown in Figure 1. It first queries

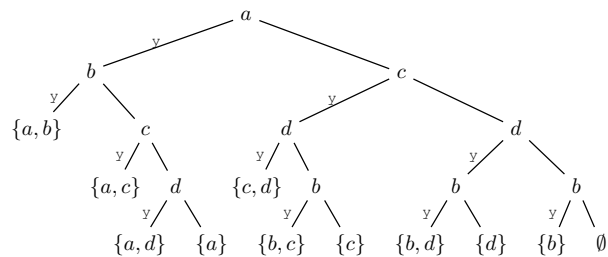


Figure 1: A tree visualizing the query policy from Example 1.

candidate a (the root of the tree); if a is available, b is queried next; otherwise, c is queried next; and so on. If, for example, $X = \{a, c, d\}$ the policy will query a , then b , and then c , and output the committee $\{a, c\}$.

In Section 4, we study query policies that query multiple candidates in parallel. Such policies can be represented by trees as well: each node will have 2^q children, where q is the number of candidates that are queried.

Approval-based committee voting rules. Let us introduce the ABC rules that are studied throughout the paper. *Thiele rules* (Thiele 1895) are parameterized by a non-increasing vector $w = (w_1, w_2, \dots)$ of non-negative real numbers. W.l.o.g. we assume that $w_1 = 1$. The rule w -Thiele returns all committees W of size k maximizing the w -Thiele score $sc_w(W) = \sum_{i \in N} \sum_{j=1}^{|A_i \cap W|} w_j$. Well-studied w -Thiele rules include Multiwinner Approval Voting (AV) with $w_{AV} = (1, 1, \dots)$, Chamberlin-Courant (CC) with $w_{CC} = (1, 0, 0, \dots)$, and Proportional Approval Voting (PAV) with $w_{PAV} = (1, \frac{1}{2}, \frac{1}{3}, \dots)$. Each w -Thiele rule is also associated with a sequential variant seq- w -Thiele. These rules start with an empty committee W and iteratively add a candidate c maximizing $sc_w(W \cup \{c\})$ until the committee contains k candidates. We refer to seq- w_{CC} -Thiele as seq-CC and to seq- w_{PAV} -Thiele as seq-PAV. Note that seq- w_{AV} -Thiele coincides with AV.

The *Method of Equal Shares* (MES) outputs all committees that can result from the following iterative procedure. Each voter $i \in N$ is assigned an initial budget of $b_i = \frac{k}{n}$. We start with an empty committee W and add candidates one at a time. Given budgets $(b_i)_{i \in N}$, we say that a candidate c is ϵ -affordable, for some $\epsilon > 0$, if $\sum_{i \in N} \min(b_i, \epsilon) = 1$. In each round, a candidate that is ϵ -affordable for the minimum ϵ is added to W and the budget b_i of each voter $i \in N_c$ is set to $b_i - \min(b_i, \epsilon)$. If no candidate is ϵ -affordable for any ϵ , MES fills the committee to size k with arbitrary candidates.

3 Safe Policies and Implementable Rules

The above definition of a query policy does not enforce the queries of the policy to be binding invitations to join the committee, i.e., the set Q_t^+ does not have to equal the output committee of the policy. Since we focus on the case of irrevocable invitations in this paper, we are interested in safe policies. A query policy is *safe* for a rule r if for each time-step t the candidate c_t is safe with regard to Q_t for r .

³For instance, for $A = (\{a, b\}, \{c\})$ and $k = 2$, many rules output $\{a, c\}$ and $\{b, c\}$: a and b are safe, but $\{a, b\}$ is not.

Definition 2. A rule is implementable if and only if there exists a safe query policy for it.⁴

To build a first intuition, we consider two well-studied voting rules, seq-PAV and PAV. We observe that the former admits a safe query policy while the latter does not.

Proposition 1. *Seq-PAV is implementable.*

Proof. Consider the query policy that in each step t queries a candidate $c_t \in C \setminus (Q_{t-1}^+ \cup Q_{t-1})$ maximizing $s_{\text{CPAV}}(Q_{t-1}^+ \cup \{c_t\})$, i.e., the candidate with the highest marginal contribution to the PAV-score of the candidates known to be available. Once $|Q_t^+| = \min(k, |X|)$ (for some t) the policy outputs Q_t^+ as the winning committee. In any round t , the queried candidate c_t is safe: if c_t is not available, it is safe by definition. If c_t is available, we can extend $Q_{t-1}^+ = Q_{t-1}^+ \cup \{c_t\}$ to a winning committee of seq-PAV because the score $s_{\text{CPAV}}(Q_{t-1}^+ \cup \{c_{t+1}\})$ only depends on the candidates Q_{t-1}^+ , which we know are available, and c_{t+1} . Thus, the policy is safe, and seq-PAV is implementable. \square

Proposition 2. *PAV is not implementable.*

Proof. Let $A = (\{c_1, c_2\}, \{c_3, c_4\}, \{c_1, c_4\}, \{c_2, c_3\})$ and $k = 2$. The committees $\{c_1, c_3\}$ and $\{c_2, c_4\}$ are selected by PAV. Consider a safe query policy. If it first queries c_1 and c_1 is available, then it must query c_3 next. If c_3 is not available, while c_2 and c_4 are, this leads to selecting a committee that does not win under PAV. By symmetry, a similar argument works if another candidate is queried first. \square

3.1 Characterizing Implementable Rules

The main point differentiating sequential-PAV from PAV in the example above seems to be the “sequentiality” of the former. In this subsection, we formalize our notion of sequentiality and show that, in essence, the existence of a safe query policy is equivalent to the rule being sequential. Intuitively, a multiwinner rule is sequential if a winning committee can be constructed by adding alternatives iteratively without depending on the approvals of other not yet selected candidates.

More specifically, we call a rule r sequential if at every round of the selection process where we have already chosen ℓ candidates c_1, \dots, c_ℓ , there is a score function f assigning a score to each unselected candidate such that choosing a candidate maximizing this function among available candidates leads to a winning committee according to the rule r . That is, it is possible to build a winning committee one candidate at a time by always picking a candidate maximizing f . Crucially, for a given candidate c , the value $f(c)$ may only depend on the target committee size, the sequence of already chosen candidates, and the approval profile restricted to the set of already chosen candidates together with c . Thus, it only depends on $(k, E, A|_{E \cup \{c\}})$, where $E = (c_1, \dots, c_\ell)$ is the sequence of already chosen candidates. We slightly abuse notation and write E for both the sequence and the set of already chosen candidates. Let $\mathcal{S}(C)$ be the set of all sequences of distinct candidates in C . Formally, a function

⁴Note that our notion of *implementability* is unrelated to the classical usage of the term for social choice functions.

$f_k: C \times \mathcal{S}(C) \times 2^C \rightarrow \mathbb{R}$ is a *marginal contribution function*, and f_k satisfies *locality* if for all E and $c \notin E$ it holds that $f_k(c, E, A) = f_k(c, E, A|_{E \cup \{c\}})$.

Definition 3. A rule r is sequential if for any committee size k , there exists a marginal contribution function f_k , satisfying locality, such that for any set of available candidates $X \subseteq C$ it holds that $\{c_1, \dots, c_k\} \in r(A, k, X)$, where for all $i \leq k$ it holds $c_i \in \max_{c \in X \setminus \{c_1, \dots, c_{i-1}\}} f_k(c, (c_1, \dots, c_{i-1}), A)$.

We omit the subscript k from the function f_k if it is clear from context. Intuitively, a rule is implementable if we can find some function f assigning a value to each unchosen candidate independently of the other unchosen candidates, such that we can pick the candidate with the maximum score and always get a committee in the output of the rule. An example of a sequential rule is MES.

Proposition 3. *MES is sequential.*

Proof. Recall that MES includes candidates one by one into the winning committee by, in each step, adding a candidate that is affordable for the minimum prize. This value depends on the remaining budgets of the voters at that point of the process. For a given k we can define a function $f_k(c, E, A)$ witnessing sequentiality for all $E, c \notin E$, and A as follows. Using the sequence of already chosen candidates E and the approval profile A , we compute the budget of each voter after the candidates in E were selected by MES. Using these budgets we compute the value b_c for which candidate c would be affordable and set $f_k(c, E, A) = \frac{1}{b_c}$. If a candidate c is not affordable given the computed budgets we set $f_k(c, E, A) = 0$. Since the computation of the budgets only depends on $A|_{E \cup \{c\}}$, the function f_k satisfies locality. Maximizing over $f_k(c, E, A)$ is equivalent to minimizing over the b_c values and thus MES is sequential with respect to these functions f_k . \square

Sequentiality is not the same as committee monotonicity (which, for the sake of simplicity, we only define for resolute rules here): A rule r is *committee monotone* if $r(A, k, X) \subseteq r(A, k+1, X)$ for all k .

Remark 1. *Sequentiality is independent of committee monotonicity. For example, MES is not committee monotone but sequential.⁵ For an example of a committee monotone rule that is not sequential, consider the following rule that we will refer to as the 2-sequence rule: Consider m candidates with a fixed order (c_1, c_2, \dots, c_m) . If more voters approve c_1 than c_2 the rule will pick the first k available candidates in the sequence (c_1, c_2, \dots, c_m) ; otherwise it will pick the first k available candidates in the reverse sequence $(c_m, c_{m-1}, \dots, c_1)$.*

Finally, for our characterization, we need one more axiom to deal with rules that behave differently based on unavailable candidates.

Definition 4. A rule r satisfies *independence of unavailable candidates (IUC)* if for all approval profiles A , sets of available candidates X , and committee sizes k it satisfies

$$r(A, k, X) = r(A', k, X)$$

⁵For a counterexample to committee monotonicity, see Proposition A.2 in the book by Lackner and Skowron (2022).

where A' denotes any approval profile obtained from A by adding or deleting approvals from candidates outside X .

Intuitively, a rule that satisfies IUC does not depend on the approvals of unavailable candidates; thus, all standard ABC rules translated to our setting satisfy this property. On the other hand, the 2-sequence rule defined above does not satisfy IUC. Note that in the definition of IUC, we do not delete the candidates in the profile A' , but only change approvals they obtained. This ensures that the profiles A and A' rely on the same set and number of candidates. We are now able to state and prove the main result of this section.

Theorem 1. *The following two statements hold:*

- (i) *Every sequential rule is implementable.*
- (ii) *Every implementable rule that satisfies IUC is sequential.*

Proof. Fix an instance (A, k) with a set of candidates C and consider any set of available candidates $X \subseteq C$. For (i), consider a sequential rule r and a marginal contribution function f , satisfying locality, witnessing the sequentiality of r . We define a safe query policy for r iteratively. For step $t \geq 1$ assume the information set $Q_t = (Q_t^+, Q_t^-)$. In the following, we slightly abuse notation and write Q_t^+ both for the set of known available candidates and for the sequence of these candidates, ordered by the time we queried them. We next query a candidate c_t maximizing $f(c_t, Q_t^+, A|_{Q_t \cup \{c_t\}})$ among candidates for which we do not know availability yet, i.e., $c_t \in \max_{c \in C \setminus Q_t \cup Q_t^-} f(c, Q_t^+, A|_{Q_t \cup \{c_t\}})$. Note that even though the query policy has no access to the set of available candidates X , due to locality, if c_t maximizes the score among all candidates in $C \setminus (Q_t^+ \cup Q_t^-)$ then it still maximizes the score among $X \setminus Q_t^+$. Thus, the committee constructed by choosing — for a sufficiently large t — all candidates in Q_t^+ as soon as $|Q_t^+| = k$ or $C = Q_t^+ \cup Q_t^-$ is in $r(A, k, X)$. Thus r is implementable.

For (ii), consider an implementable rule r satisfying IUC. There exists a safe query policy for r . We consider the steps of the query policy iteratively and also define our function f inductively: For $\ell \in \{0, 1, \dots, k-1\}$ let $E = (c_1, \dots, c_\ell)$ be the sequence of already chosen candidates by the policy. Note that the case $\ell = 0$ covers the initial step where E is the empty sequence. By induction, we assume that f is already defined for all prefixes of E .

For a possible candidate $c \in C$ we say that c is choosable if $f(c, (c_1, \dots, c_{t-1}), A) < f(c_t, (c_1, \dots, c_{t-1}), A)$ for all $t \in [\ell]$, i.e., in each previous turn, c got a lower score than the chosen candidates. If $\ell = 0$, all candidates are still choosable. Let C' be the set of all choosable candidates.

After E was chosen by r 's query policy, let c'_1 be the candidate that would be queried next by the query policy. We set the score $f(c'_1, E, A|_{E \cup \{c'_1\}}) = |C'|$. Next, inductively, for $i \geq 1$, assume that the candidates c'_1, \dots, c'_i are unavailable and let c'_{i+1} be the candidate queried next by r 's policy.

We set $f(c'_{i+1}, E, A|_{E \cup \{c'_{i+1}\}}) = f(c'_i, E, A|_{E \cup \{c'_i\}}) - 1$. We further set $f(c', E, A') = f(c'_1, E, A|_{E \cup \{c'_1\}})$ for all c' and $A' \subseteq A$. All other (not choosable) candidates will not appear in the set over which the scores are maximized, and therefore we can set their score to be -1 .

Rule	SEQ	IMPL	IUC	CM
AV				
seq-PAV				
PAV	X	X		X
MES				X
2-sequence	X		X	
2-sequence \cup AV			X	

Table 1: An overview of some rules' compliance with the axioms discussed in this section. *CM* stands for committee monotonicity. The 2-sequence rule is introduced in Remark 1.

To show that r is sequential with regard to f let X be the set of available candidates. Now again, let c_1, \dots, c_ℓ be the first ℓ candidates picked by f . We inductively show that these are also the ℓ candidates that the safe query policy would query in this instance. Now let $c_{\ell+1}$ be the candidate who is queried as the $(\ell+1)$ -th candidate by the query policy. Since $c_{\ell+1}$ was not queried before, we know that $f(c_{\ell+1}, (c_1, \dots, c_{t-1}), A) < f(c_t, (c_1, \dots, c_{t-1}), A)$ for all $t \in [\ell]$. Further, we know that for any other $c \in X$, due to IUC, this $c \in X$ would not be queried before $c_{\ell+1}$ even if other candidates from C' were available. Hence, $c_{\ell+1}$ would receive a higher score than any other candidate in X ; therefore, f and the query policy agree. Further, we need to show that f satisfies locality. For this, we observe that for each set of chosen candidates E the set of choosable candidates from C is the same. Thus, the score assigned to c is always the same based on these candidates; it follows that f satisfies locality. \square

The IUC property in Part (ii) of Theorem 1 is necessary: The 2-sequence rule defined in Remark 1, which fails IUC, is implementable but not sequential. Moreover, sequentiality does not imply IUC: the rule that outputs all AV committees plus the committee chosen by the 2-sequence rule is sequential (because AV is) but fails IUC.

The following characterization result follows immediately from Theorem 1.

Corollary 1. *Let r be a rule satisfying IUC. Then, r is sequential if and only if it is implementable.*

By Proposition 3 and Theorem 1, MES is implementable. The same holds for all sequential Thiele rules and many other sequential rules such as Phragmén's sequential rule (Phragmén 1894) and the maximin support method (Sánchez-Fernández et al. 2021). For an overview of some rules and their compliance with the axioms discussed in this section, see Table 1.

3.2 Finding Safe Candidates for Thiele Rules

Theorem 1 implies that safe candidates always exist for sequential Thiele rules. Moreover, it is easy to find such a safe candidate: simply take the candidate in $C \setminus (Q^+ \cup Q^-)$ with the highest marginal contribution w.r.t. the Thiele score. On the other hand, we show that it is computationally intractable to decide whether a given candidate is safe.

Theorem 2. For any sequential and non-sequential Thiele rule except AV, even in the first step of the query policy, it is coNP-hard to decide whether a given candidate is safe.

For non-sequential Thiele rules it may happen that no candidate is safe. Moreover, we show that it is computationally intractable to decide whether there exists a safe candidate or not in a given instance.

Theorem 3. For any w -Thiele rule except AV, it is coNP-hard to decide whether a safe candidate exists.

4 Parallel Queries

In many scenarios it is desirable or even necessary to speed up the process of querying candidates. One example is invitations of speakers for a conference: the timespan between planning the conference and the actual starting date does often not allow to ask every single invited speaker only after the previous one has accepted or declined the offer to give a talk. Motivated by these scenarios, we study parallel queries where we ask multiple candidates at once, allowing us to run the querying process in parallel. Query policies are easily generalized to parallel queries. The difference to the definition in Section 2 is that if q candidates are queried in parallel at a given node, then this node has 2^q children corresponding to all availability configurations of these candidates. As a technicality, whenever we speak about having a safe set of size q , we assume that there are at least q candidates that have not been queried so far.

Example 2. Consider 8 voters, 4 candidates a, b, c, d , and $k = 2$; 4 voters approve $\{a, b\}$, 3 voters approve $\{a, c\}$, and one voter approves $\{d\}$. Under seq-PAV, the set $\{a, b\}$ is safe: the winning committee is $\{a, b\}$ if both a and b are available, $\{b, c\}$ if $X = \{b, c, d\}$, and $\{a, c\}$ if $X = \{a, c, d\}$.

On the other hand, under seq-CC, there is no safe set of size 2. It would have to be $\{a, d\}$ since this is the winning set if all candidates are available; but if $X = \{b, c, d\}$, then $\{b, c\}$ wins and thus $\{a, d\}$ is not safe (and d is not even safe as a singleton).

4.1 Parallel Queries for Thiele Rules

It is clear that under AV, finding a maximal number of safe candidates is easy.

Proposition 4. For AV, if $k - k'$ safe candidates are already queried, querying the k' candidates with the next highest approval score is always safe.

The (easy) proof uses the fact that for any $k' \leq k$, if x is among the best available k' candidates when the set of available candidates is X , then it remains among the best available k' candidates if the set of available candidates is $X' \subset X$ (and $x \in X'$).

It turns out that we cannot do the same for other sequential Thiele rules, as there might be instances where only one safe candidate exists.

Proposition 5. For any sequential Thiele rule other than AV, there are instances where the maximum size of a safe set is 1.

In the proof we construct an instance similar to the one in Example 2 (but depending on the weight vector w), which

shows that for a sequential w -Thiele rule, the case where no safe set of size at least 2 exists can occur in round $q - 1$, where q is the number of weights in w equal to 1. It is also easy to see that it cannot occur earlier: if w starts with q consecutive 1's, then the rule is equivalent to AV for the first q candidates that get selected, and thus a safe set of size q exists if no candidates have been queried yet. These are exactly the q candidates with the highest approval score. Thus, it is natural to ask whether a given instance admits more than q safe candidates. It turns out that this problem is computationally intractable.

Theorem 4. Given a w -Thiele rule with $w_1 = \dots = w_q = 1$ and $w_{q+1} < 1$, it is coNP-hard to decide whether $q + 1$ safe candidates exist if no candidates have been queried yet.

4.2 Parallel Queries for Proportionality Axioms

Finally, we turn to investigating parallel queries for proportionality axioms. For this, we introduce two commonly used proportionality notions: *Proportional Justified Representation (PJR)* and *Extended Justified Representation (EJR)* (Sánchez-Fernández et al. 2017; Aziz et al. 2017).

Definition 5. Given an ABC instance (A, k) , a group $N' \subseteq N$ of voters is ℓ -cohesive if $|N'| \geq \frac{\ell n}{k}$ and $|\bigcap_{i \in N'} A_i| \geq \ell$. A committee W of size k satisfies PJR if for each ℓ -cohesive group N' it holds that $|\bigcup_{i \in N'} A_i \cap W| \geq \ell$. A committee W of size k satisfies EJR if for each ℓ -cohesive group N' there is some voter $i \in N'$ with $|A_i \cap W| \geq \ell$.

We are interested in whether these axioms are implementable, i.e., whether committees satisfying these axioms can be found with a safe query protocol.⁶ Since MES satisfies EJR (and thus also PJR) (Peters and Skowron 2020), we know from Proposition 3 and Theorem 1 that PJR and EJR are both implementable: We can always find a safe candidate to add such that we will end up with a committee satisfying the axiom in the end.

We now investigate whether we can query multiple candidates at once. We focus our attention on the first query, i.e., we assume that no candidate has been queried yet. We show that there always exists a safe query consisting of k candidates for PJR. To do this, we introduce a variant of priceability (Peters and Skowron 2020).

Definition 6 (Priceability with individual budgets). A committee W satisfies priceability with individual budgets if and only if there are individual budgets $B_i \geq 0$ and payment functions $p_i: C \rightarrow [0, B_i]$ for every voter $i \in N$ such that

C1 If $p_i(c) > 0$ then $c \in A_i$ for any $c \in C$ and $i \in N$,

C2 If $p_i(c) > 0$ then $c \in W$ for any $c \in C$ and $i \in N$,

C3 $\sum_{c \in C} p_i(c) \leq B_i$ for any $i \in N$,

C4 $\sum_{i \in N} p_i(c) = 1$ for any $c \in W$, and

C5 $\sum_{i \in N} B_i - \sum_{c' \in W} p_i(c') \leq 1$ for any $c \notin W$.

We call the pair $(B, p) = (B_i)_{i \in N}, (p_i)_{i \in N}$ a price system and say that W is supported by this price system.

⁶Formally, safeness for an axiom can be defined as safeness for the voting rule that outputs all committees satisfying the axiom.

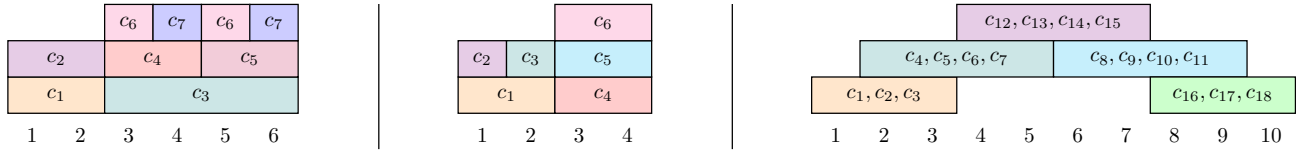


Figure 2: Instances discussed in Examples 3 and 4. Voters correspond to integers and approve all candidates placed above them.

It is easy to see that price systems satisfying **C1-C4** are equivalent to *load distributions* on which Phragmén’s rules are based (Brill et al. 2017). Hence, we can define *leximax-Phragmén* as the rule which outputs all priceable committees that are supported by a sorted vector $B = (B_i)_{i \in N}$ of budgets that is lexicographically minimal.⁷

Example 3. Consider the instance depicted on the left in Figure 2 and let $k = 3$. This instance admits several *leximax-Phragmén* committees, in which every voter gets a budget of $B_i = \frac{1}{2}$. This is optimal, since at least one voter must get a budget of $\frac{k}{n} = \frac{1}{2}$. One such committee is $\{c_1, c_4, c_5\}$ with the first two voters paying for c_1 , the second two for c_4 , and the last two for c_5 . This committee is safe for PJR: independent of which subset X of candidates is available, we can extend $\{c_1, c_4, c_5\} \cap X$ to a committee satisfying PJR. For instance, if c_4 was unavailable, we could include c_3 instead.

However, in this instance we can also see that not every PJR committee is safe for PJR: For example, the committee $\{c_1, c_2, c_3\}$ satisfies PJR, but if c_3 was unavailable, $\{c_1, c_2\}$ could not be extended to a committee satisfying PJR since the groups $\{3, 4\}$ and $\{5, 6\}$ form 1-cohesive groups.

As a side note, we observe that the committee $\{c_1, c_4, c_5\}$ selected by *leximax-Phragmén* is not safe for *leximax-Phragmén* itself. For instance, if c_3 and c_4 were unavailable, the only two *leximax-Phragmén* committees are $\{c_1, c_6, c_7\}$ and $\{c_2, c_6, c_7\}$.

We can show that every committee selected by *leximax-Phragmén* constitutes a safe set of candidates (of size k) for PJR. In the proof of Theorem 5, we also show how the query protocol can be “completed” using queries of size 1.

Theorem 5. Assume that no candidate has been queried yet. Then, every *leximax-Phragmén* committee is safe for PJR.

For EJR, however, the situation turns out to be trickier. First, we show that neither PAV nor MES always select k candidates that are safe for EJR.

Example 4. For PAV, consider the instance depicted in the center of Figure 2. Here, for $k = 4$, PAV selects (among other committees) $\{c_1, c_4, c_5, c_6\}$. But if c_1 is unavailable, both c_2 and c_3 need to be included in the committee to satisfy EJR (since voters 1 and 2 are each 1-cohesive on their own).

For MES, consider the instance depicted on the right in Figure 2 and let $k = 10$. The committee $\{c_1, c_2, c_3, c_{12}, \dots, c_{18}\}$ is selected by MES (among other committees). However, if c_{12} is unavailable, to satisfy EJR, we need to include a candidate from c_4, \dots, c_7 and a candidate from c_8, \dots, c_{11} to satisfy the 4-cohesive voter groups approving these candidates.

⁷Since *leximax-Phragmén* satisfies **C5** in addition to **C1-C4**, we can assume that we are optimizing over **C1-C5**.

On the bright side, we can at least improve upon the initial query size of 1 and show that, in the first step, there always exists a safe set of two candidates for EJR.

Theorem 6. Assume that no candidate has been queried yet and that $k \geq 2$. Then, there exists a set of size at least two that is safe for EJR.

Proof. Let c_1 be an approval winner and c_2 be an approval “runner-up,” i.e., $n_2 := |N_{c_2}| \geq |N_c|$ for all $c \in C \setminus \{c_1\}$ and $n_1 := |N_{c_1}| \geq n_2$. We distinguish two cases.

Case 1: $n_2 \geq \frac{2n}{k}$. Then, $n_1 \geq \frac{2n}{k}$ as well. We show that there is a safe set of two candidates for MES, which is therefore also safe for EJR. Under MES, c_1 is bought first and the agents in N_{c_2} have at least $\frac{k}{2n}$ of their budget left. Thus, c_2 is still $\frac{k}{2n}$ -affordable. Further, any other candidate c is at most $\frac{1}{|N|}$ -affordable and thus at most $\frac{k}{2n}$ -affordable. Hence, c_2 can still be bought by MES even if c_1 was bought (and especially if c_1 was unavailable). Therefore, independent of availability, the set $\{c_1, c_2\}$ can be safely queried.

Case 2: $n_2 < \frac{2n}{k}$. Then, no 2-cohesive group exists. Thus, EJR only needs to satisfy 1-cohesive groups and is equivalent to PJR. The statement now follows from Theorem 5. \square

5 Discussion

We have initiated the study of multiwinner voting with possibly unavailable candidates. Our key assumptions were approval ballots and binding queries. Also, we do not have availability probabilities. Moving away from these assumptions leads to a variety of interesting directions for future work. We discuss three directions in turn.

It would be a natural next step to consider multiwinner voting based on ordinal preferences and study how rules such as single transferable vote (STV) can be adapted to deal with possibly unavailable candidates. We maintain that our characterization of implementable rules (Theorem 1) holds for that setting as well, as the underlying logic of the proof is independent of the ballot format.

If queries were non-binding, there would be a trivial query policy: query all candidates and select the desired committee. But the task would become nontrivial again if queries had costs as in the paper by Boutilier et al. (2014).

In some settings, we might have probabilistic information on the availability of candidates. For non-implementable rules such as PAV, we might want to look for binding query policies that maximize the *expected* score of the committee.

Within the setting considered in this paper, it is open whether our guarantee for PJR (Theorem 5) can also be achieved by a computationally tractable rule, and whether our guarantee for EJR (Theorem 6) can be improved.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant BR 4744/2-1 and the Graduiertenkolleg “Facets of Complexity” (GRK 2434), by the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), and by a Humboldt Research Award of the Alexander von Humboldt Foundation.

References

- Aziz, H.; Brill, M.; Conitzer, V.; Elkind, E.; Freeman, R.; and Walsh, T. 2017. Justified Representation in Approval-Based Committee Voting. *Social Choice and Welfare*, 48(2): 461–485.
- Baldiga, K. A.; and Green, J. R. 2013. Assent-maximizing social choice. *Social Choice and Welfare*, 40(2): 439–460.
- Barrot, N.; Gourvès, L.; Lang, J.; Monnot, J.; and Ries, B. 2013. Possible Winners in Approval Voting. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*, 57–70. Springer-Verlag.
- Boutilier, C.; Lang, J.; Oren, J.; and Palacios, H. 2014. Robust Winners and Winner Determination Policies under Candidate Uncertainty. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 1391–1397. AAAI Press.
- Boutilier, C.; and Rosenschein, J. S. 2016. Incomplete Information and Communication in Voting. In *Handbook of Computational Social Choice*, 223–258. Cambridge University Press.
- Brill, M.; Freeman, R.; Janson, S.; and Lackner, M. 2017. Phragmén’s Voting Methods and Justified Representation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 406–413. AAAI Press.
- Do, V.; Hervouin, M.; Lang, J.; and Skowron, P. 2022. Online Approval Committee Elections. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 251–257.
- Flanigan, B.; Gözl, P.; Gupta, A.; Hennig, B.; and Procaccia, A. 2021. Fair algorithms for selecting citizens’ assemblies. *Nature*, 596: 548–552.
- Freeman, R.; Kahng, A.; and Pennock, D. M. 2020. Proportionality in Approval-Based Elections With a Variable Number of Winners. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, 132–138. ijcai.org.
- Grivet Sébert, A.; Maudet, N.; Perny, P.; and Viappiani, P. 2021. Preference Aggregation in the Generalised Unavailable Candidate Model. In *Proceedings of the 7th International Conference on Algorithmic Decision Theory (ADT)*, 35–50. Springer.
- Imber, A.; Israel, J.; Brill, M.; and Kimelfeld, B. 2022. Approval-Based Committee Voting under Incomplete Information. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 5076–5083. AAAI Press.
- Lackner, M.; and Skowron, P. 2022. *Multi-Winner Voting with Approval Preferences*. Springer.
- Lu, T.; and Boutilier, C. E. 2010. The unavailable candidate model: a decision-theoretic view of social choice. In *Proceedings of the 11th ACM Conference on Electronic Commerce (ACM-EC)*, 263–274.
- Oren, J.; Filmus, Y.; and Boutilier, C. 2013. Efficient vote elicitation under candidate uncertainty. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 309–316.
- Peters, D.; and Skowron, P. 2020. Proportionality and the Limits of Welfarism. In *Proceedings of the 21st ACM Conference on Economics and Computation (ACM-EC)*, 793–794. ACM Press. Full version arXiv:1911.11747 [cs.GT].
- Phragmén, E. 1894. Sur une méthode nouvelle pour réaliser, dans les élections, la représentation proportionnelle des partis. *Öfversigt af Kongliga Vetenskaps-Akademiens Förhandlingar*, 51(3): 133–137.
- Sánchez-Fernández, L.; Elkind, E.; Lackner, M.; Fernández, N.; Fisteus, J. A.; Basanta Val, P.; and Skowron, P. 2017. Proportional justified representation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 670–676. AAAI Press.
- Sánchez-Fernández, L.; Fernández, N.; Fisteus, J. A.; and Brill, M. 2021. The Maximin Support Method: An Extension of the D’Hondt Method to Approval-Based Multiwinner Elections. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 5690–5697. AAAI Press.
- Terzopoulou, Z.; Karpov, A.; and Obratzsova, S. 2021. Restricted domains of dichotomous preferences with possibly incomplete information. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 5726–5733. AAAI Press.
- Thiele, T. N. 1895. Om Flerfold Valg. *Oversigt over det Kongelige Danske Videnskabernes Selskabs Fordhandlingar*.