

Diverse Committees with Incomplete or Inaccurate Approval Ballots

Feline Lindeboom¹, Martijn Brehm², Davide Grossi^{1,2}, Pradeep Murukannaiah³

¹University of Groningen, ²University of Amsterdam, ³TU Delft

Abstract

We study diversity in approval-based committee elections with incomplete or inaccurate information. We define diversity according to the Maximum Coverage problem, which is known to be NP-complete, with a best attainable polynomial time approximation ratio of $1 - 1/e$. In the incomplete information setting, voters vote only on a small portion of the candidates, and we prove that getting arbitrarily close to the optimal approximation ratio w.h.p. requires $\Omega(m^2)$ non-adaptive queries, where m is the number of candidates. This motivates studying adaptive querying algorithms, that can adapt their querying strategy to information obtained from previous query outcomes. In that setting, we lower this bound to only $\Omega(m)$ queries. We propose a greedy algorithm to match this lower bound up to log-factors. We prove the same $\tilde{\Theta}(m)$ bound for the generalized problem of Max Cover over a matroid constraint, using a local search algorithm. Specifying a matroid of valid committees lets us implement extra structural requirements on the committee, like quota. In the inaccurate information setting, voters' responses are corrupted with a small probability. We prove $\tilde{\Theta}(nm)$ queries are required to attain a $(1 - 1/e)$ -approximation with high probability, where n is the number of voters. While the proven bounds show that all our algorithms are viable asymptotically, they also show that some of them would still require large numbers of queries in instances of practical relevance. Using real data from Polis as well as synthetic data, we observe that our algorithms perform well also on smaller instances, both with incomplete and inaccurate information.

1 Introduction

Consensus has grown among political scientists, democracy practitioners, and decision makers alike, that effective involvement of citizens in policy decisions should be regarded as a high priority for democratic institutions at all levels, from local, to national, to regional [19, 23]. Currently, it is on digital democracy platforms especially that research and development efforts are concentrating [5, 13], in order to provide citizens with effective civic participation tools [27, 24].

A wealth of these types of digital democracy tools have been developed and deployed around the world in the last decade: LiquidFeedback [2], Consul, YourPriorities, Decidim, Polis [31], to mention a few.¹ In different forms, all these applications allow users to provide free-text input to public deliberations and enable them to express their own opinions on the input of others. Supporting such processes involves the use of algorithms for information-processing problems. One such problem concerns how to effectively summarize the current state of the deliberation: how does ‘the group’ currently think about the issues being deliberated upon?

One approach to this problem, which is followed for instance in Polis, consists of selecting sets of statements proposed by users, based on how much support the statements have elicited from other users. As [14] already noticed, this approach can be conceptualized as an approval-based multi-winner voting problem [9, 18], as studied in computational social choice [4]: users ‘vote’ for other users’ statements by approving or disapproving them. Unlike in standard multi-winner approval voting, however, information about who supports which statements is now sparse, as no user can possibly

¹See www.liquidfeedback.org; www.consulproject.org; www.citizens.is; www.decidim.org; www.polis.is

express whether they agree or disagree with every single statement contributed by their peers. For the same reason, voters can become inaccurate in representing their true beliefs, when having to answer many questions. Such a deliberation summarization problem can thus be thought of as an approval-based multi-winner voting problem in which ballots are incomplete or inaccurate, or both. From this perspective, [14] focused on the problem of designing selection algorithms yielding summarizations that meet specific forms of proportional representation proposed in the computational social choice literature [1], while querying users’ opinions as efficiently as possible.

Our work builds on the approach put forth in [14], but focuses on the property of diversity instead. Diversity has received less attention than proportional representation in the multi-winner voting literature [18]. Yet, we argue that diversity is especially important for deliberation summarization. Firstly, diverse summarizations can guarantee higher inclusivity, an attribute one could desire in and of itself [24]. Secondly, a diverse summary gives a broad, informative view of ‘the group’s’ opinion. This can be valuable especially in deliberation summarization contexts, where the number of candidates may be very large. Thirdly, cognitive diversity is argued, e.g., by [19], to also be of epistemic value from a crowd-wisdom perspective. This is relevant because the selected statements typically serve as a basis for further discussions. We use the established formalization of diversity in multi-winner voting based on the Chamberlin-Courant score [7] and the Maximum Coverage problem [25, 30]. Our focus thus lies on the computational problem of constructing summarizations that maximize the Chamberlin-Courant score, while querying users’ opinions as efficiently as possible.

2 Related work and contribution

Diversity in multi-winner voting and Max Cover In multi-winner voting, the diversity of a committee is commonly quantified by the *Chamberlin-Courant* score [7]. In *approval-based* multi-winner voting, a voter’s representative in a committee W then is any elected candidate they approve of, and the approval-based Chamberlin-Courant (going forward: Chamberlin-Courant) score of W equals the number of voters that have a representative in W . [28] proved that the winner determination problem based on the Chamberlin-Courant score is NP-complete. Portraying candidates by their set of approving voters, the problem amounts to selecting a set of candidates that maximizes the ‘coverage’ of the set of voters. This one-to-one correspondence between Chamberlin-Courant score-based diversity in approval-based multi-winner voting, and the unit-weight Maximum Coverage problem in combinatorial optimization, was first studied in [30].

An instance of the unit-weight Maximum Coverage problem, Max Cover going forward, consists of a set V of n elements v_i , a family $C \subseteq \mathcal{P}(V)$ of m subsets c_j , and a number k . The goal is to select k subsets so that $|\cup_{j=1}^k c_j|$ is maximized. [15] proved, by reduction from Set Cover, that the decision problem formulation of Max Cover is NP-complete, and that a greedy algorithm is $(1 - 1/e)$ -approximate. Shortly after, [10], using a reduction from approximating Max 3SAT-5, proved that for any $\epsilon > 0$, Max Cover cannot be approximated in polynomial time within a ratio of $1 - 1/e + \epsilon$, unless $P = NP$; the simple greedy algorithm is thus optimal. Many years later, [8] showed that under the Gap Exponential Time Hypothesis,² for any $\epsilon > 0$, there exists no FPT-approximation algorithm for parameter k that approximates Max Cover within factor $1 - 1/e + \epsilon$. Essentially, this shows a brute-force approach going over all possible size- k subsets is the best we can do if we insist on obtaining an approximation ratio better than $1 - 1/e$.

Finally, a generalization of the problem was studied by [11], who present a local search algorithm that is $(1 - 1/e)$ -approximate for Max Cover over a matroid constraint, where the greedy algorithm strands at ratio $1/2$ in this generalized setting.

²For $k \geq 3$, define $s_k := \inf\{\delta : \exists \text{ an algorithm that solves } k\text{-SAT in } 2^{\delta n} \text{ time}\}$. ETH states that for $k \geq 3$, $s_k > 0$; k -SAT cannot be solved in subexponential time, for $k \geq 3$. ETH was first formulated by [16] and would imply that $P \neq NP$.

Incomplete		Inaccurate
Adaptive		$\tilde{\Theta}(nm)$
Matroid	No matroid	
$\tilde{\Theta}(m)$	$\tilde{\Theta}(m)$	$\Omega(m^2)$

Table 1: Overview of query complexity bounds in m and n .

Incomplete or inaccurate information Single-winner voting with incomplete information has received some attention over the years in the computational social choice literature, see [3]. More recently, [14] studied proportionality in approval-based committee elections under incomplete information. The paper presents a version of local search PAV that queries voters and, using $O(mk^6 \log k \log m)$ queries, finds a solution that satisfies Extended Justified Representation [1] and Optimal Average Satisfaction (based on the notion of average satisfaction, introduced by [33]) with high probability. The result extends to an α -approximate version of both axioms, in which case the query complexity decreases by a factor k^3 . The same study also presents a lower bound of $\Omega(m^{11})$ for the query complexity of non-adaptive algorithms, that can not adapt their querying strategy information obtained from previous queries. In the field of fair division of indivisible items, [21] study the number of queries required to find a round-robin allocation. When voters' responses are corrupted with a small probability, the query complexity is in $\Omega(nm + m \log m)$ and $O(nm \log m)$.

Our contribution We study diversity in approval-based committee elections when information elicited from voters is incomplete or inaccurate. We measure diversity of a solution with its Max Cover, or equivalently, Chamberlin-Courant, score. For a large part, our work combines the two lines of research outlined above, and is based in particular on [11], [14] and [21]. We make three main contributions. *First*, in a setting with incomplete information, we prove that getting arbitrarily close to the optimal approximation ratio w.h.p. requires $\Omega(m^2)$ non-adaptive queries (Appendix C). This motivates studying *adaptive* querying algorithms, that can adapt their querying strategy to information obtained from previous query outcomes. In that setting, we lower this bound to only $\Omega(m)$ queries. We adapt the greedy algorithm to match this lower bound up to log-factors (Theorem 2). We prove the same $\tilde{\Theta}(m)$ bound for the generalized problem of Max Cover over a matroid constraint (Theorem 6). Specifying a matroid of valid committees lets us implement external diversity requirements, like quota. *Second*, in the inaccurate information setting, we prove that recovering the optimal approximation ratio w.h.p. requires $\tilde{\Theta}(nm)$ queries (Theorem 8). Despite these positive asymptotic bounds, summarized in Table 1, our results do involve sizeable constant overheads for some of our algorithms, which make viability on real-world instance questionable. So, *third*, using real data from Polis and synthetic data that we produced by adapting methods from [32] to our setting, we empirically show that our algorithms perform considerably better than our worst-case analysis suggests. Importantly, this appears to hold even in situations in which votes are both incomplete *and* inaccurate.

3 Preliminaries

An instance of an approval-based committee election problem consists of a set V of n voters, a set C of m candidates, and a natural number k . The goal is to elect a committee $W \subseteq C$ of size $k \leq m$, based on the opinions of the voters in V . We assume, based on observations on online deliberation summarization platforms, that $m \in \Theta(n)$ and $k \ll m$. An instance also contains approval information: every voter $v_i \in V$ approves of a subset of the candidates. Usually, this approval information is known upfront and expressed in the form of an *approval set* $A(i)$ for each voter $v_i \in V$. The sequence of sets $A = (A(1), \dots, A(n))$ is called the *approval profile*, and we call this setting the *perfect information*

setting. This paper studies incomplete and inaccurate information settings; the respective models are detailed in the corresponding section. To measure diversity of an elected committee, we use the Chamberlin-Courant score.

Definition 1 (Chamberlin and Courant 1983). *On any approval-based committee election instance (V, C, A, k) , the Chamberlin-Courant (CC) score of a committee $W \subseteq C$, $|W| = k$ is*

$$CC(W) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{A(i) \cap W \neq \emptyset\}}(i). \quad (1)$$

In the Chamberlin-Courant decision problem, the input is an instance (V, C, A, k) and an integer x , and the question is whether it is possible to achieve score $\frac{x}{n}$ using k candidates. In the optimization problem, the input is an instance (V, C, A, k) and the task is to maximize the score using k candidates.

We will use the terms Max Cover and Chamberlin-Courant interchangeably. We do the same for voters/elements, to candidates/sets, and committees/covers.

4 Incomplete information

In the *incomplete information setting*, the complete approval profile $A = (A(1), \dots, A(n))$ is replaced by a function $A(i, j) := \mathbb{1}_{\{c_j \in A(i)\}}(i, j)$ which equals 1 in case voter v_i approves candidate c_j and 0 otherwise. This is a generalization of the original model, as querying all voters on all candidates recovers the complete approval profile. As such, by making nm queries to A we can find a $(1 - 1/e)$ -approximate solution, e.g. with a greedy approach. In Appendix C, we show that any non-adaptive querying algorithm must make $\Omega(m^2)$ queries to A to get arbitrarily close to this optimal approximation ratio w.h.p., where non-adaptive means that the sequence of queries made must be specified beforehand. In our setting, $m \in \Theta(n)$, which means $\Omega(m^2) = \Omega(nm)$, making no improvement over the complete information setting. We can do better studying adaptive querying algorithms, that can adapt their querying strategy to information obtained from previous query outcomes. For such adaptive algorithms we will show that obtaining an approximation ratio arbitrarily close to this optimal ratio w.h.p. requires only $\tilde{O}(m)$ queries to A .

To see that we need at least $\Omega(m)$ queries, imagine an instance where only one candidate has any approvals. Any algorithm must select this candidate to attain a positive approximation ratio. In the worst-case, finding this voter w.h.p. requires sampling voters for each of the m candidates. In this section we study algorithms that witness this lower bound $\Omega(m)$. We first adapt a greedy algorithm for Max Cover and then do the same for a local search algorithm for the generalized problem of optimizing over a matroid constraint. Both algorithms achieve an approximation ratio arbitrarily close to the optimal ratio w.h.p. using $\tilde{O}(m)$ queries, matching the lower bound (up to log-factors).

4.1 Upper bound for the unconstrained setting

The standard greedy algorithm We prepare the ground by considering the greedy algorithm for the perfect information setting, Algorithm 1, which achieves the optimal approximation ratio of $1 - 1/e$ [15]. Algorithm 1 elects, in each iteration, the candidate that yields the largest immediate increase in Max Cover score. We write $\Delta(W, c)$ for the increase in Max Cover score obtained by adding candidate c to committee W . We give a novel proof of its approximation ratio in Appendix A.

The greedy query algorithm We adapt Algorithm 1 to the incomplete information setting. Our strategy is to repeatedly sample sets of voters (of size $\ell \leq n$) uniformly at random, in order to estimate $\Delta(W, c)$ for all $c \notin W$. The algorithm then behaves like the standard greedy algorithm: iteratively

selecting the candidate c maximizing this estimate of $\Delta(W, c)$. By carefully bounding the deviation from the true value $\Delta(W, c)$, we can guarantee that our algorithm attains a ratio arbitrarily close to the optimal approximation ratio.

To write this more formally, note that if we ask *all* voters for their votes on some query set $Q \subseteq C$ (of size t), the responses allow us to, for any set $S \subseteq Q$, compute

$$p_S := \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{A(i) \cap S \neq \emptyset\}}(i). \quad (2)$$

The value p_S equals the Chamberlin-Courant score of S , and in this case $\Delta(W, c) = p_{W \cup \{c\}} - p_W$, so we can compute $\Delta(W, c)$ with complete information on a query set containing $W \cup \{c\}$. Sampling ℓ voters uniformly at random and querying them $Q \subseteq C$, we can compute, for every subset $S \subseteq Q$,

$$\hat{p}_S := \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbb{1}_{\{A(i) \cap S \neq \emptyset\}}(i), \quad (3)$$

that estimates p_S . We write $\hat{\Delta}(W, c) := \hat{p}_{W \cup \{c\}} - \hat{p}_W$ for an estimate of $\Delta(W, c)$ based on $\hat{p}$. Using this approach, we obtain Algorithm 2.

We assume that $\sum_{v_i \in V} \mathbb{1}_{\{|A(i)| \geq 1\}}(i) \geq k$; there exist at least k voters with a non-empty approval set. We call instances for which this is true, *non-trivial*; a trivial instance would attain maximal score simply by electing a candidate for each voter v_i for which $A(i) \cap C \neq \emptyset$. Theorem 2 gives an upper bound on the number of queries required to run Algorithm 2 on non-trivial instances.

Theorem 2. *Let an instance be non-trivial, $\delta > 0$, $\gamma \in (0, 1)$, and $k < t \leq m$. Then, with probability at least $1 - \delta$, Algorithm 2 is $(1 - 1/e)\gamma$ -approximate for Max Cover with query complexity $O\left(\left(\frac{\gamma}{1-\gamma}\right)^2 km \log\left(\frac{km}{\delta}\right)\right) \in O_{\delta, \gamma, k}(m \log m)$.*

Importantly, the query complexity does not depend on n . Asymptotically, $O(m \log m)$ is much smaller than querying the entire ballot of size nm . The constant overhead, which increases with k and γ and decreases with δ , is quite small. For instance, if $m = 1000$, $\gamma = 0.85$, $\delta = 0.05$, $k = 8$ and $t = 20$, Theorem 2 requires 432 920 queries. Starting with $n > 432$ voters this will be smaller than nm .

4.2 Upper bound for a setting with matroid constraints

Matroids in multi-winner voting We can imagine situations where we want to place restrictions on the set of possible committees to be elected. For instance, perhaps we want to include extra diversity requirements, such as ‘the committee must contain at least x candidates of category Y ’. In practice, x would be a natural number and Y would be a category like a gender, race, etc. In online deliberation settings, these attributes would extend to users. For such purposes, we make a selection of which committees are “allowed” to be elected, and collect them in a set $\mathcal{M}$. This is fundamentally different from what we have so far meant with diversity, which was defined in terms of the approval profile. It is not obvious that any algorithm adapts well to this restriction. Fortunately, when $\mathcal{M}$ is a matroid, we can mend this problem [11].

ALGORITHM 1: GREEDY

Input: Numbers $n, m, k \in \mathbb{N}$ with $k \leq m$, set V of n voters, set C of m candidates.

Output: Committee $W \subseteq C$ of size k .

Let $W = \{\}$ be an empty set;

for $i = 1, \dots, k$, **do**

Add $c' \in \operatorname{argmax}_{c \notin W} \Delta(W, c)$ to W .

end

ALGORITHM 2: GREEDY-INCOMPLETE

Input: Numbers $n, m, k \in \mathbb{N}$ with $k \leq m$, set V of n voters, set C of m candidates, query size t with $m \geq t > k$, and $\gamma \in (0, 1)$, $\delta > 0$.

Output: Committee $W \subseteq C$ of size k .

Set $\epsilon = \frac{(1-\gamma)e}{\gamma(e-1)}$ and $\ell = \lceil \frac{2}{\epsilon^2} (\log(\frac{2mk}{\delta})) \rceil$;

Let $W = \{\}$ be an empty set;

for $i = 1, \dots, k$, **do**

1. Construct the smallest set of query sets $\mathcal{Q} = \{Q_i\}_i$ with $Q_i \subseteq C$ and $|Q_i| = t$ for all i , and such that $W \subseteq \bigcap \mathcal{Q}$ and $\bigcup \mathcal{Q} = C$, and present each query set to ℓ voters sampled uniformly at random.
2. For all $c \notin W$, determine $\hat{\Delta}(W, c)$ as an estimate of $\Delta(W, c)$ using ℓ voters and query set Q containing $W \cup c$.
3. Add $c' \in \operatorname{argmax}_{c \notin W} \hat{\Delta}(W, c)$ to W .

end

Definition 3 (matroid). *A matroid $\mathcal{M}$ over a set C is a selection of subsets (called independent sets) with*

1. $\emptyset \in \mathcal{M}$,
2. if $A \in \mathcal{M}$ and $B \subset A$ then $B \in \mathcal{M}$, and
3. for all $A, B \in \mathcal{M}$ with $|A| > |B|$, there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{M}$.

Property 3) guarantees that all maximal independent sets have the same cardinality. We call such sets *bases*, and their common cardinality is called the *rank* of the matroid.

Example 4 (uniform matroid). *Consider $\mathcal{M} = \{W \subseteq C : |W| \leq k\}$. This is the uniform matroid of rank k : 1) $|\emptyset| = 0 \leq k$, 2) for any set $W \in \mathcal{M}$ with $|W| = j \leq k$, for any subset $W' \subseteq W$, $|W'| \leq |W| \leq k$, and 3) for any two feasible sets $W, W' \in \mathcal{M}$ with $|W'| < |W|$, we will have, for any $c \in W \setminus W'$, that $|W' \cup \{c\}| \leq k$.*

Thus, taking for $\mathcal{M}$ the uniform matroid of rank k , we retrieve the original problem: the problem with matroid constraints is a generalization of the original problem.

Example 5 (matroid implementing quota). *For $X \cup Y = C$, $X \cap Y = \emptyset$, define a set of bases $M = \{W \subseteq C : |W| = k, |W \cap Y| = |W \cap X| = \frac{1}{2}k\}$. Then $\mathcal{M} = M \cup \{V : V \subseteq W, W \in M\}$ is a matroid of rank k .*

In Example 5, only the bases comply with the quota, but this is not a problem. When an algorithm outputs $A \in \mathcal{M}$ with $|A| = \ell < k$, then by requirement 3) in Definition 3, using any basis element $B = \{b_1, \dots, b_k\} \in \mathcal{M}$, we can make a path $A, A \cup \{b_1\}, A \cup \{b_1\} \cup \{b_2\}, \dots, A \cup \{b_1\} \cup \dots \cup \{b_{k-\ell}\} =: A'$ from A to attain a basis element $A' \in \mathcal{M}$ that contains A (and has a score at least as high).

The standard non-oblivious local search algorithm As before, we start with the perfect information setting. A classical (or oblivious) local search algorithm for Max Cover starts with an arbitrary solution and, in each iteration, swaps one or multiple of the elected sets for unelected sets in order to improve the objective function value. It thus *searches* in the *local* neighborhood of the solution. It stops when no local improvement is possible. At worst, this occurs after all exponentially many options have been exhausted. With the use of approximate local search, this running time problem can be resolved at a small cost in the approximation ratio, see also [26]. In practice, this means we terminate the algorithm when we encounter an improvement smaller than some parameter $\beta > 0$ ³.

³Using a partial enumeration technique, we could even eliminate this small cost again, see [6].

ALGORITHM 3: LOCAL SEARCH- β

Input: Numbers $n, m, k \in \mathbb{N}$ with $k \leq m$, set V of n voters, set C of m candidates, $\beta > 0$.

Output: Committee $W \subseteq C$, of size k .

Choose $W \subseteq C$ such that $|W| = k$, and $c \in W$ and $c' \notin W$ so that $(W \cup \{c'\}) \setminus \{c\} \in \mathcal{M}$;

repeat

1. $W = (W \cup \{c'\}) \setminus \{c\}$.
2. Let $\mathcal{E}$ be the set of all valid exchanges for W according to $\mathcal{M}$.
3. Pick $(c', c) \in \operatorname{argmax}_{(x,y) \in \mathcal{E}} \Delta(W, x, y)$.

until $\Delta(W, c', c) \leq \beta$;

Over a matroid constraint, the greedy algorithm attains only approximation ratio $\frac{1}{2} < 1 - 1/e$. The approximation ratio of oblivious local search is $\frac{k-1}{2k-\ell-1}$ when ℓ sets are exchanged in each iteration [11]. For $k = 2$ and $\ell = 1$, this also equals $\frac{1}{2}$. Thus, we switch to a non-oblivious local search algorithm that uses an auxiliary objective function f for the iterative procedure. The function adds temporary weight to elements covered multiple times, the idea being that elements covered multiple times, remain covered after the next exchange. Such a function can thus prevent getting stuck in bad local optima. We denote by α_j the temporary weight associated to an element covered j times, and write $h_i(W)$ for the amount of times that v_i is covered by W . The auxiliary objective function is defined

$$f(W) = \frac{1}{n} \sum_{i=1}^n \alpha_{h_i(W)}. \quad (4)$$

Note that, with $\alpha_0 = 0$ and $\alpha_j = 1$ for all $j > 0$, we retain the original oblivious objective function, but if we set $\alpha_j > \alpha_1$ for $j > 1$, we add additional weight to elements covered multiple times. We define

$$\alpha_0 = 0, \quad \alpha_1 = 1 - \frac{1}{e}, \quad \alpha_{j+1} = (j+1)\alpha_j - j\alpha_{j-1} - \frac{1}{e}.$$

This choice for the sequence $(\alpha_n)_{n \in \mathbb{N}_0}$ is optimal and, for any $\gamma \in (0, 1)$, with this objective function, the non-oblivious local search algorithm, with parameter β (decreasing in γ) is $(1 - 1/e - \gamma)$ -approximate and runs in polynomial time [11].

We adjust the non-oblivious local search algorithm of [11] to suit our setting better, see Algorithm 3, where for $c \in W$ and $c' \notin W$, we now write $\Delta(W, c', c) := f((W \cup \{c'\}) \setminus \{c\}) - f(W)$. We elaborate on the adaptations in Appendix B, but most importantly, the changes do not affect the approximation guarantees of the algorithm: for $\beta = C_1 \frac{\gamma}{k \log k}$, Algorithm 3 is $(1 - 1/e - \gamma)$ -approximate for any $\gamma \in (0, 1)$ and some universal constant $C_1 \leq \frac{\log i}{\alpha_i(1-1/e-\gamma)}$ for any $i \leq k$ [11, Corollary 6].

The non-oblivious local search query algorithm We turn to the incomplete information setting. Recall that $A(i, j)$ equals 1 when voter v_i approves of candidate c_j and 0 otherwise, and that we write $A(i)$ for the set of candidates voter v_i approves. Given a query set $Q \subseteq C$, for any $S \subseteq Q$, we redefine

$$p_S := \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{A(i) \cap Q = S\}}(i) \quad (5)$$

as the probability that a voter, of all the candidates in Q , approves of exactly the set S . This is different from how we defined p_S for Algorithm 2: Algorithm 2 requires, for any voter v_i , knowing just whether they have a representative in set $S \subseteq Q$, whereas Algorithm 4 needs the number of representatives.

ALGORITHM 4: LS- β -INCOMPLETE

Input: Numbers $n, m, k \in \mathbb{N}$ with $k \leq m$, set V of n voters, set C of m candidates, $\beta > 0$, query size t , with $m \geq t > k$, constants $\delta > 0$ and $\xi \geq 1$.

Output: Committee $W \subseteq C$, of size k .

$$\epsilon = \frac{\xi-1}{2\xi} \cdot \beta, \ell = \left\lceil \frac{(2-2/e)^2}{2\epsilon^2} \log \left(\frac{2 \cdot (m-k)k \cdot \xi \alpha_k}{\delta \cdot \beta} \right) \right\rceil;$$

Choose $W \subseteq C$ such that $|W| = k$, and $c \in W$ and $c' \notin W$ so that $(W \cup \{c'\}) \setminus \{c\} \in \mathcal{M}$;

repeat

1. $W = (W \cup \{c'\}) \setminus \{c\}$;
2. Let $\mathcal{E}$ be the set of all valid exchanges for W according to $\mathcal{M}$.
3. Construct the smallest set of query sets $\mathcal{Q} = \{Q_i\}_i$ with $Q_i \subseteq C$ and $|Q_i| = t$ for all i , and such that $W \subseteq \bigcap \mathcal{Q}$ and $\bigcup \mathcal{Q} = C$, and present each query set to ℓ voters sampled uniformly at random.
4. For all $c \in W$, $c' \notin W$, determine $\hat{\Delta}(W, c', c)$ as an estimate of $\Delta(W, c', c)$ using ℓ voters and query set $\mathcal{Q}$ containing $W \cup \{c'\}$.
5. Pick $(c', c) \in \arg\max_{(x,y) \in \mathcal{E}} \hat{\Delta}(W, x, y)$.

until $\hat{\Delta}(W, c', c) < \beta - \epsilon$;

With p_S for $S \subseteq Q$, we can compute $f(Q)$:

$$\begin{aligned} f(Q) &= \frac{1}{n} \sum_{i=1}^n \alpha_{h_i(Q)} = \frac{1}{n} \sum_{i=1}^n \sum_{S \subseteq W} \mathbb{1}_{\{A(i) \cap Q = S\}}(i) \cdot \alpha_{|S|} \\ &= \sum_{S \subseteq Q} \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{A(i) \cap Q = S\}}(i) \cdot \alpha_{|S|} = \sum_{S \subseteq Q} p_S \cdot \alpha_{|S|}. \end{aligned}$$

We can compute $\Delta(W, c', c)$ with complete information on a query set containing $W \cup \{c'\}$. When information is incomplete, we estimate: sampling $\ell \leq n$ voters uniformly at random and presenting them with query set $Q \subseteq C$, we can compute, for every subset $S \subseteq Q$,

$$\hat{p}_S := \frac{1}{\ell} \sum_{i=1}^{\ell} \mathbb{1}_{\{A(i) \cap Q = S\}}(i), \quad (6)$$

that estimates p_S . We write $\hat{\Delta}(W, c', c) = \hat{f}((W \cup \{c'\}) \setminus \{c\}) - \hat{f}(W) = \sum_{S \subseteq (W \cup \{c'\}) \setminus \{c\}} \hat{p}_S \cdot \alpha_{|S|} - \sum_{S \subseteq W} \hat{p}_S \cdot \alpha_{|S|}$ for an estimate of $\Delta(W, c', c)$ based on the values $\hat{p}$. Algorithm 4 is our local search query algorithm. Theorem 6 gives the number of queries required for Algorithm 4. As expected, this is higher than stated in Theorem 2.

Theorem 6. Let $|c_j| \geq 1 \forall j$, $\delta > 0$, $\gamma \in (0, 1)$, $m \geq t > k \geq 3$. Fix $\beta = C_2 \frac{1-\gamma}{\gamma k \log k}$ for some constant C_2 . Then, w.p. at least $1 - \delta$, Algorithm 4 is $(1 - 1/e)\gamma$ -approximate for Max Cover with query complexity $O\left(\left(\frac{\gamma k \log k}{1-\gamma}\right)^3 m \log\left(\frac{m \gamma k^2 \log k}{\delta(1-\gamma)}\right)\right) \in O_{\delta, \gamma, k}(m \log m)$.

Like with GREEDY-INCOMPLETE, the query complexity does not depend on n , and is asymptotically much smaller than querying the entire ballot nm . Unlike before, there is sizeable constant overhead increasing with k and γ , and decreasing with δ . Especially the dependence in k is much stronger than before. For the same example, $m = 1000$, $\gamma = 0.85$, $\delta = 0.05$, $k = 8$ and $t = 20$, Theorem 6 requires $9.52180 \cdot 10^{11}$ queries. Only for $n > 9.52180 \cdot 10^8$ this is smaller than nm .

5 Inaccurate information

In the inaccurate information setting, voters do hand in complete ballots, but the voters are inaccurate in reporting their true approvals: we assume each query $A(i, j)$ is incorrect with a probability p . We model this by defining, for $p \in (0, \frac{1}{2})$, a p -inaccurate query as $A_p(i, j) := A(i, j) \oplus X$, where $X \sim \text{Bernoulli}(p)$ ⁴. Samples of X are independent between voters, between candidates, and between consecutive draws of the same query. If $p = 0$, we would have accurate information, and with nm queries we would find a $(1 - 1/e)$ -approximate solution (e.g. using the greedy algorithm). When answers may be corrupted, we can simply pose every question multiple times to compensate for the uncertainty. We do this to obtain an upper bound on the number of queries required to still acquire a $(1 - 1/e)$ -approximate solution with high probability, adapting Algorithm 1, attaining Algorithm 6 (see Appendix D). Moreover, we provide a matching lower bound (up to log-factors). Both results hold for the problem of optimizing over a matroid constraint.

Theorem 7. *Let $p \in (0, \frac{1}{2})$, $\delta > 0$, $n, m \in \mathbb{N}$. Then there exists an algorithm that is $(1 - 1/e)$ -approximate for Max Cover in the p -inaccurate query model w.p. at least $1 - \delta$ and with query complexity $O(nm \log(nm/\delta))$.*

Theorem 8. *Let $p \in (0, \frac{1}{2})$, $\delta > 0$, $n, m \in \mathbb{N}$. Then any algorithm that is $(1 - 1/e)$ -approximate for Max Cover in the p -inaccurate query model w.p. at least $1 - \delta$ has expected query complexity $\Omega(nm \log(1/\delta))$.*

The two bounds match (up to log-factors). Unlike before, we access the entire ballot, making the bound depend on n .

6 Experiments

Motivation We run our algorithms on real-life data and synthetic data with realistic structure.⁵ First, we establish empirically that the greedy and local search algorithms with complete information (Algorithm 1 and 3) consistently achieve a higher Max Cover score than the standard APPROVAL VOTING, electing the k most approved candidates, justifying the study of our algorithms in the first place. Second, although we have proven that our querying algorithms can overcome inaccurate and incomplete voter responses with constant overhead, this constant overhead can still be very large. We show that in practice our algorithms beat the optimal approximation ratio, even when using much fewer queries than theoretically required, and even when we combine inaccurate and incomplete information (a setting that we did not study theoretically).

Data We test our algorithms on 18 open-use datasets provided by Polis⁶. We did multiple pre-processing steps, most importantly completing the Polis data. As anticipated, participating voters typically vote on only a small portion of the statements, so we obtain a sparse comment-participant-matrix. Since these discussions have been completed already, we are not able to query voters anymore, so, for the sake of running our querying algorithms, we need to artificially complete the data. For details on this and other pre-processing steps we implemented, we refer to Appendix E.

To obtain more data beyond the 18 Polis datasets, we generate synthetic data that are structurally similar to the Polis data. Specifically, we employ the (q, ϕ) -resampling model to sample approval elections, introduced by [32]. In this model, $q \in [0, 1]$ represents the fraction of approvals and $\phi \in [0, 1]$ represents the spread of approvals, so that $\phi = 0$ means all voters are identical and approve the exact same $\lfloor q \cdot m \rfloor$

⁴This is essentially the classification noise model from PAC learning [17] applied to our setting.

⁵All code and data is available at <https://github.com/martijnmartijnmartijnmartijn/Diverse-Committees-with-Incomplete-or-Inaccurate-Approval-Ballots>

⁶<https://github.com/compdemocracy/openData>

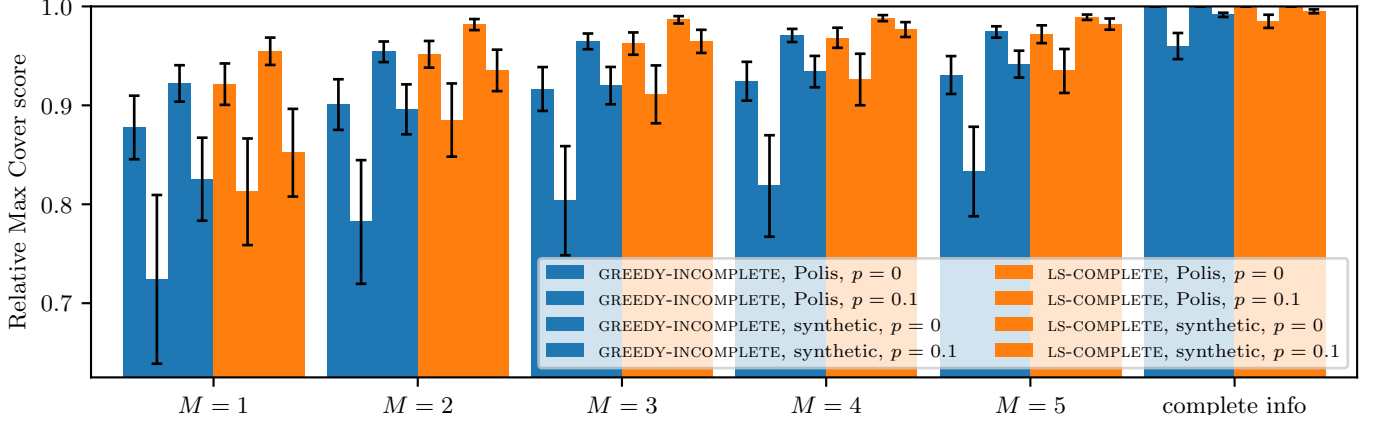


Figure 1: Max Cover score of GREEDY-INCOMPLETE (Algorithm 2) and LS-INCOMPLETE (Algorithm 4) on the 18 Polis and 100 synthetic data sets for $k = 8$. The algorithms are configured so that each voter is expected to be queried only $M = 1, 2, 3, 4, 5$ times. This was done for $p = 0$ (accurate responses) and $p = 0.1$ (inaccurate responses). On each of the 118 data sets, we ran 50 random trials of the algorithms, plotting the mean and standard deviation. All scores shown are relative to the complete and accurate information setting.

	APPROVAL	LOCAL SEARCH	GREEDY
Polis	0.77 ± 0.071	0.80 ± 0.074	0.82 ± 0.085
Synthetic	0.98 ± 0.0040	0.99 ± 0.0029	0.99 ± 0.0019

Table 2: Mean and standard deviation of the Max Cover score of the algorithms GREEDY, LOCAL SEARCH (Algorithms 2 and 4) and APPROVAL VOTING with complete information on the 18 Polis datasets and 100 synthetic datasets. We do 20 random trials of LOCAL SEARCH.

candidates, while $\phi = 1$ means each candidate is approved with probability q independently so that the spread of approvals is maximal. After pre-processing the Polis data, we find $q = 0.0891$ and $\phi = 0.693$. Appendix E contains an explanation of how to arrive at these values. We sampled 100 elections according to the $(0.0891, 0.693)$ -resampling model with $n = 1000$ and $m = 400$, as $m/n = 0.4$ on average for the Polis datasets, and the average number of voters is roughly 1000.

Results To answer our first question, we ran GREEDY (Algorithm 1), LOCAL SEARCH- β (Algorithm 3)⁷ and APPROVAL VOTING on the 118 data sets, including 20 random trials for each dataset for Algorithm 3, because of the random starting committee. Table 2 shows the mean Max Cover scores of the algorithms over all runs, separately for Polis data and synthetic data. On Polis data the mean score of all three algorithms is around 0.8, while on synthetic data the scores are nearly 1. GREEDY performs better than LOCAL SEARCH, which in turn performs better than APPROVAL VOTING. For Polis data, this difference in scores lies within the standard deviation. However, we stress that on no iteration of *any* Polis dataset, APPROVAL VOTING ‘beat’ the other two algorithms. The standard deviation thus reflects that different Polis datasets admit different Max Cover scores (across all three algorithms). On average over all Polis data, GREEDY and LOCAL SEARCH score higher than APPROVAL VOTING by a factor of 1.063 ($\sigma = 0.038$) and by a factor of 1.045 ($\sigma = 0.032$), respectively. On the synthetic data, GREEDY always produced a committee with a higher Max Cover score than APPROVAL VOTING, while LOCAL SEARCH did so on 95 out of 100 datasets (on 4 datasets it lost to APPROVAL VOTING on 1 out of 50 random trials; on 1 dataset it lost on 6 out of 50 trials), but the score was still within a factor of 0.996 of the score of APPROVAL VOTING. On average over all synthetic data, GREEDY and LOCAL SEARCH score higher than APPROVAL VOTING by a factor of 1.013 ($\sigma = 0.004$) and by a factor of 1.008 ($\sigma = 0.004$), respectively. We therefore conclude that our two algorithms, and especially GREEDY, consistently outperform APPROVAL VOTING by

⁷We took β as in Theorem 6 and write LOCAL SEARCH going forward

a decent factor.

To answer our second question, we take the scores of GREEDY and LOCAL SEARCH from Table 2 as our baseline and inspect how close we can get to these scores when information is incomplete and/or inaccurate, using a realistic number of queries. For the incomplete information model, taking $t = 20$, we write M for the expected number of query sets of size t that each voter is presented with. We then configure GREEDY-INCOMPLETE (Algorithm 2) and LS-INCOMPLETE (Algorithm 4) so that $M = 1, 2, 3, 4, 5$, which is much lower than what would theoretically be required. We do 50 random trials of each algorithm (due to the random querying of voters). For the inaccurate information model, we ignore the theoretically required repetitions of the queries, and run the standard greedy and local search algorithms, but with $p = 0.1$. Finally, we combine both of the above by running Algorithm 2 and Algorithm 4 with the above values of M but for $p = 0.1$. See figure 1 for the Max Cover scores obtained in these experiments.

We see that both algorithms with accurate responses ($p = 0$) obtain a score very close (0.85-0.95) to versions of the algorithms with complete information, even when $M = 1$. With complete information, both algorithms obtained an average Max Cover score of around 0.8. Multiplying by 0.9 still yields a score above the worst-case approximation ratio of $1 - 1/e \approx 0.63$, even when the optimal solution would have a Max Cover score of 1. Example 9 highlights the great performance witnessed in the experiments for $p = 0$.

Example 9. Figure 1 shows that for $M = 5$, the Max Cover score attained by LS-INCOMPLETE, on average across 18 Polis datasets with 50 runs per set, is approximately 0.95 times the score attained by LOCAL SEARCH. The dataset *vtaiwan.uberx* has $m = 197$ and $n = 1921$. For these values of m and n , taking $\gamma = 0.95$, $k = 8$, $t = 20$, and $\delta = 0.05$, we would need $M = 7.109 \cdot 10^8$.

With $p = 0.1$, but with complete information, the performance decreases by a factor of 0.95-0.97 on average, which yields a score confidently above the worst-case approximation ratio of $1 - 1/e$. As a comparison: taking again dataset *vtaiwan.uberx* with $\delta = 0.05$ would require repeating each query 32 times, but here we do not repeat queries at all.

Taking both $p = 0.1$ and incomplete information, not meeting the theoretically required number of queries for *either* scenario, the performance takes a noticeable hit. The algorithms perform worse than the versions with complete information by a factor of 0.7-0.8 for $M = 1$ increasing to 0.85-0.95 for $M = 5$. This again implies that we are above the worst-case approximation ratio. Note that the standard deviation can become relatively large here, especially for the Polis data, which are more varied than the synthetic data.

7 Outlook

Our work is the first to address diversity in approval-based committee elections in the context of online civic participation platforms. Measuring diversity by the Chamberlin-Courant score, we prove diverse committees can be found by querying only a small fraction of the voters, even when responses may be inaccurate. This remains true in the presence of external diversity constraints on the committee, such as quota. Our algorithms match lower bounds on the query complexity (up to log-factors). We verify these theoretical results empirically on both real-life and synthetic data.

Our results open up several directions for future research. *First*, it would be interesting to combine the incomplete and inaccurate information models explicitly in theory. *Second*, it would be desirable to lift our results on inaccurate information to richer error models, e.g., where the error probability changes over time and/or between voters. *Third*, by viewing whom to query on what as an optimization problem—‘the comment routing problem’ by [31]—we hope to lower the query complexity further. *Fourth*, we intend to explore the significance of our results for applications in active learning [29], to optimize the querying of a group of annotators in distributed data labelling tasks (see, e.g., [22]).

Acknowledgments This research was partially supported by Hybrid Intelligence Center (<https://www.hybrid-intelligence-centre.nl/>), a 10-year programme funded by the Netherlands Organisation for Scientific Research, under Grant No. 024.004.022.

Contact details

Feline Lindeboom
University of Groningen
Groningen, The Netherlands
Email: felinelineboom@proton.me

Martijn Brehm
University of Amsterdam
Amsterdam, The Netherlands
Email: m.a.brehm@uva.nl

Davide Grossi
University of Groningen
Groningen, The Netherlands
Email: d.grossi@rug.nl

Pradeep Murukannaiah
TU Delft
Delft, The Netherlands
Email: p.k.murukannaiah@tudelft.nl

References

- [1] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. Justified representation in approval-based committee voting. *Social Choice and Welfare*, 48(2):461–485, 2017.
- [2] Jan Behrens, Axel Kistner, Andreas Nitsche, and Björn Swierczek. *The Principles of LiquidFeedback*. Interaktive Demokratie, 2014.
- [3] Craig Boutilier and Jeffrey S Rosenschein. Incomplete information and communication in voting. *Handbook of computational social choice*, 10:223–257, 2016.
- [4] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.
- [5] Markus Brill. Interactive democracy. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1183–1187, 2018.
- [6] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a Monotone Submodular Function Subject to a Matroid Constraint. *SIAM Journal on Computing*, 40(6): 1740–1766, January 2011. ISSN 0097-5397, 1095-7111. doi: 10.1137/080733991. URL <http://epubs.siam.org/doi/10.1137/080733991>.
- [7] 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, September 1983. ISSN 0003-0554, 1537-5943. doi: 10.2307/1957270.
- [8] Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT approximations for k-median and k-means. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 42:1–42:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi: 10.4230/LIPICS.ICALP.2019.42. URL <https://doi.org/10.4230/LIPIcs.ICALP.2019.42>.
 - [9] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. Multiwinner voting: A new challenge for social choice theory. *Trends in computational social choice*, 74(2017):27–47, 2017.
 - [10] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, July 1998. ISSN 0004-5411, 1557-735X. doi: 10.1145/285055.285059.
 - [11] Yuval Filmus and Justin Ward. The Power of Local Search: Maximum Coverage over a Matroid. *LIPIcs, Volume 14, STACS 2012*, 14:601–612, 2013. ISSN 1868-8969. doi: 10.4230/LIPICS.STACS.2012.601.
 - [12] L Gordon. Tutorial on large deviations for the binomial distribution. *Bulletin of Mathematical Biology*, 51(1):125–131, 1989.
 - [13] Davide Grossi, Ulrike Hahn, Michael Mäs, Andreas Nitsche, Jan Behrens, Niclas Boehmer, Markus Brill, Ulle Endriss, Umberto Grandi, Adrian Haret, et al. Enabling the digital democratic revival: A research program for digital democracy. *arXiv preprint arXiv:2401.16863*, 2024.
 - [14] Daniel Halpern, Gregory Kehne, Ariel D. Procaccia, Jamie Tucker-Foltz, and Manuel Wüthrich. Representation with Incomplete Votes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(5):5657–5664, June 2023. doi: 10.1609/aaai.v37i5.25702.
 - [15] Dorit S. Hochbaum and Anu Pathria. Analysis of the greedy approach in problems of maximumk-coverage. *Naval Research Logistics*, 45(6):615–627, September 1998. ISSN 0894-069X, 1520-6750. doi: 10.1002/(SICI)1520-6750(199809)45:6<615::AID-NAV5>3.0.CO;2-5.
 - [16] Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, March 2001. ISSN 00220000. doi: 10.1006/jcss.2000.1727. URL <https://linkinghub.elsevier.com/retrieve/pii/S0022000000917276>.
 - [17] Michael J Kearns and Umesh Vazirani. *An introduction to computational learning theory*. MIT press, 1994.
 - [18] Martin Lackner and Piotr Skowron. *Multi-Winner Voting with Approval Preferences*. SpringerBriefs in Intelligent Systems. Springer International Publishing, Cham, 2023. ISBN 978-3-031-09015-8 978-3-031-09016-5. doi: 10.1007/978-3-031-09016-5.
 - [19] Hélène Landemore. *Democratic Reason: Politics, Collective Intelligence, and the Rule of the Many*. Princeton University Press, Princeton, NJ Oxford, first paperback printing edition, 2017. ISBN 978-0-691-17639-0 978-0-691-15565-4.
 - [20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 1 edition, July 2020. ISBN 978-1-108-57140-1 978-1-108-48682-8. doi: 10.1017/9781108571401.
 - [21] Zihan Li, Pasin Manurangsi, Jonathan Scarlett, and Warut Suksompong. Complexity of Round-Robin Allocation with Potentially Noisy Queries. In Guido Schäfer and Carmine Ventre, editors, *Algorithmic Game Theory*, volume 15156, pages 520–537. Springer Nature Switzerland, Cham, 2024. ISBN 978-3-031-71032-2 978-3-031-71033-9. doi: 10.1007/978-3-031-71033-9_29.
 - [22] Enrico Liscio, Luciano C Siebert, Catholijn M Jonker, and Pradeep K Murukannaiah. Value preferences estimation and disambiguation in hybrid participatory systems. *arXiv preprint arXiv:2402.16751*, 2024.
 - [23] John G Matsusaka. Let the people rule. In *Let the People Rule*. Princeton University Press, 2020.
 - [24] Anna Mikhaylovskaya. Enhancing deliberation with digital democratic innovations. *Philosophy & Technology*, 37(1):3, 2024.
 - [25] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, December 1978. ISSN

0025-5610, 1436-4646. doi: 10.1007/BF01588971.

- [26] James B. Orlin, Abraham P. Punnen, and Andreas S. Schulz. Approximate Local Search in Combinatorial Optimization. *SIAM Journal on Computing*, 33(5):1201–1214, January 2004. ISSN 0097-5397, 1095-7111. doi: 10.1137/S0097539703431007. URL <http://epubs.siam.org/doi/10.1137/S0097539703431007>.
- [27] Ismael Peña-López et al. Innovative citizen participation and new democratic institutions: Catching the deliberative wave. OECD Publishing, 2020.
- [28] Ariel D. Procaccia, Jeffrey S. Rosenschein, and Aviv Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, April 2008. ISSN 0176-1714, 1432-217X. doi: 10.1007/s00355-007-0235-2.
- [29] Burr Settles. Active learning literature survey. 2009.
- [30] Piotr Skowron and Piotr Faliszewski. Fully Proportional Representation with Approval Ballots: Approximating the MaxCover Problem with Bounded Frequencies in FPT Time. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), February 2015. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v29i1.9432.
- [31] Christopher Small, Michael Bjorkegren, Timo Erkkilä, Lynette Shaw, and Colin Megill. Polis: Scaling deliberation by mapping high dimensional opinion spaces. *Recerca: Revista de Pensament i Anàlisi*, 26(2), 2021.
- [32] Stanisław Szufa, Piotr Faliszewski, Lukasz Janeczko, Martin Lackner, Arkadii Slinko, Krzysztof Sornat, and Nimrod Talmon. How to sample approval elections? In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 496–502. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/71. URL <https://doi.org/10.24963/ijcai.2022/71>. Main Track.
- [33] Luis Sánchez-Fernández, Edith Elkind, Martin Lackner, Norberto Fernández, Jesús Fisteus, Pablo Basanta Val, and Piotr Skowron. Proportional justified representation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. doi: 10.1609/aaai.v31i1.10611. URL <https://ojs.aaai.org/index.php/AAAI/article/view/10611>.

A Proof of Theorem 2

We prove that GREEDY-INCOMPLETE (Algorithm 2) can get arbitrarily close to the optimal approximation ratio for Max Cover using $O(m \log m)$ queries to voters. This result was stated as Theorem 2. We start by giving a new proof that the regular greedy algorithm (Algorithm 1) achieves the optimal approximation ratio of $1 - \frac{1}{e}$, and we adapt this proof for Lemma 13, and consequently for the main result, Theorem 2.

Lemma 10. *Algorithm 1 is $(1 - \frac{1}{e})$ -approximate for the maximum coverage problem, and runs in time $O(kmn)$.*

For the proof we use the following notation: for $i \leq k$, let GR_i denote the increase in Max Cover score the greedy algorithm attains in iteration i , and $\text{GR}(i)$ for the Max Cover score of the greedy solution after i iterations. This means $\text{GR}(k)$ is the score after termination. We write $\text{OPT}(k)$ for the score an optimal solution, consisting of k candidates, would have.

Proof. We start with the approximation ratio. Like [15, Lemma 1], we first prove that, for $\ell \leq k$, $\text{GR}_\ell \geq (\text{OPT}(k) - \text{GR}(\ell - 1))/k$. Fix an optimal solution. At the start of iteration ℓ , the difference between the greedy score and the optimal score is $\text{OPT}(k) - \text{GR}(\ell - 1)$. Thus, there are *at least* this many elements covered in the (fixed) optimal solution, that have not been covered yet by GREEDY up to that point (it could be more: the set selected by GREEDY at the start of iteration ℓ , needn't be a subset of the optimal solution). These $\text{OPT}(k) - \text{GR}(\ell - 1)$ elements are covered by the k sets in the optimal solution so, by the pigeon hole principle, there exists a set in the optimal solution, that is not currently selected by GREEDY, that will cover at least $(\text{OPT}(k) - \text{GR}(\ell - 1))/k$ of these elements. Then, in iteration ℓ , the greedy algorithm picks the set that covers the most additional elements, so these must be at least $(\text{OPT}(k) - \text{GR}(\ell - 1))/k$ many. We use this relation to obtain an expression for $\text{GR}(k)$ in terms of $\text{OPT}(k)$. For all k ,

$$\begin{aligned}
\text{GR}(k) &= \text{GR}(k-1) + \text{GR}_k \\
&\geq \text{GR}(k-1) + \frac{\text{OPT}(k) - \text{GR}(k-1)}{k} \\
&= \frac{\text{OPT}(k)}{k} + \text{GR}(k-1) \left(1 - \frac{1}{k}\right) \\
&\geq \frac{\text{OPT}(k)}{k} + \left(\frac{\text{OPT}(k)}{k} + \text{GR}(k-2) \left(1 - \frac{1}{k}\right)\right) \left(1 - \frac{1}{k}\right) \\
&\geq \frac{\text{OPT}(k)}{k} + \left(\frac{\text{OPT}(k)}{k} + \left(\frac{\text{OPT}(k)}{k} + \text{GR}(k-3) \left(1 - \frac{1}{k}\right)\right) \left(1 - \frac{1}{k}\right)\right) \left(1 - \frac{1}{k}\right) \\
&\dots \\
&\geq \frac{\text{OPT}(k)}{k} + \frac{\text{OPT}(k)}{k} \left(1 - \frac{1}{k}\right) + \frac{\text{OPT}(k)}{k} \left(1 - \frac{1}{k}\right)^2 + \dots + \frac{\text{OPT}(k)}{k} \left(1 - \frac{1}{k}\right)^{k-2} + \text{GR}(1) \left(1 - \frac{1}{k}\right)^{k-1} \\
&\geq \frac{\text{OPT}(k)}{k} \left(\sum_{i=0}^{k-2} \left(1 - \frac{1}{k}\right)^i + \left(1 - \frac{1}{k}\right)^{k-1}\right) \\
&= \frac{\text{OPT}(k)}{k} \sum_{i=0}^{k-1} \left(1 - \frac{1}{k}\right)^i \\
&= \frac{\text{OPT}(k)}{k} \left(k \left(1 - \left(1 - \frac{1}{k}\right)^k\right)\right) \\
&= \text{OPT}(k) \left(1 - \left(1 - \frac{1}{k}\right)^k\right).
\end{aligned}$$

Because $\left(1 - \frac{1}{k}\right)^k$ is an increasing sequence with $\lim_{k \rightarrow \infty} \left(1 - \frac{1}{k}\right)^k = \frac{1}{e}$, we can bound the last expression from below by $\text{OPT}(k) \left(1 - \frac{1}{e}\right)$, which proves the approximation ratio.

Regarding the time complexity, the algorithm chooses k candidates sequentially. For each of these k steps, we have to go over all m candidates to check which adds most to the current score. Per candidate, we have to check for each of the n voters whether it is already represented (we can keep track of the represented voters in one set). This gives a time complexity in $O(kmn)$. $\square$

Remark 11. The convergence goes rather quickly; for $k = 10$, we already have $1 - \left(1 - \frac{1}{k}\right)^k = 1 - \frac{9}{10}^{10} \approx 0.651$, whereas $1 - 1/e \approx 0.632$. However, for small committee sizes the actual worst case approximation ratio can be substantially smaller than the general ratio of $1 - 1/e$: when $k = 3$, the ratio is actually $1 - \frac{2}{3}^3 \approx 0.704$.

Proposition 12. On any non-trivial instance, $\text{GR}(k) \geq k$.

Proof. We proceed by contraposition. Assume $\text{GR}(k) < k$, which is equivalent to $\text{GR}(k) \leq k-1$, so (by the pigeonhole principle) there exists $i \leq k$ such that $\text{GR}_i = 0$. Because $\text{GR}_i \leq \text{GR}_{i-1}$ for all $i \leq k$, we must have (at least) $\text{GR}_k = 0$; no candidate with a yet uncovered voter is left in iteration k , since otherwise the greedy algorithm would pick that over an empty candidate and then $\text{GR}_k > 0$. At this point, $\text{GR}(k-1) \leq k-1$ and no uncovered voters are left, so $\sum_{v_i \in V} \mathbb{1}_{\{|A(i)| \geq 1\}}(i) \leq k-1$, which means the instance is trivial. $\square$

For the incomplete information setting we need to estimate approval information of the total population based on the information of only few voters. This can lead to errors. Write $\text{GR}-\epsilon$ for any adaptation of Algorithm 1 that, in each iteration, chooses any candidate that yields the largest increase in score *up to* $-\epsilon$; see Algorithm 5 for an example. We obtain the following approximation.

Lemma 13. For any non-trivial instance, any $\text{GREEDY-}\epsilon$ algorithm is $\left(\left(1 - \frac{1}{e}\right) \cdot \left(\frac{1}{1 + \epsilon(1 - \frac{1}{e})}\right) \right) -$ approximate.

For $i \leq k$, define $\text{GR-}\epsilon_i$ and $\text{GR-}\epsilon(i)$ similarly to before.

Proof. We write out $\text{GR-}\epsilon(k)$ recursively like before, but now subtracting ϵ in each iteration. For any k , we have

$$\begin{aligned}
\text{GR-}\epsilon(k) &= \text{GR-}\epsilon(k-1) + \text{GR-}\epsilon_k \\
&\geq \text{GR-}\epsilon(k-1) + \frac{\text{OPT}(k) - \text{GR-}\epsilon(k-1)}{k} - \epsilon \\
&= \frac{\text{OPT}(k)}{k} + \text{GR-}\epsilon(k-1) \left(1 - \frac{1}{k}\right) - \epsilon \\
&\geq \frac{\text{OPT}(k)}{k} + \left(\frac{\text{OPT}(k)}{k} + \text{GR-}\epsilon(k-2) \left(1 - \frac{1}{k}\right) - \epsilon \right) \left(1 - \frac{1}{k}\right) - \epsilon \\
&= \frac{\text{OPT}(k)}{k} - \epsilon + \left(\frac{\text{OPT}(k)}{k} - \epsilon \right) \left(1 - \frac{1}{k}\right) + \text{GR-}\epsilon(k-2) \left(1 - \frac{1}{k}\right)^2 \\
&\geq \frac{\text{OPT}(k)}{k} - \epsilon + \left(\frac{\text{OPT}(k)}{k} - \epsilon \right) \left(1 - \frac{1}{k}\right) + \left(\frac{\text{OPT}(k)}{k} - \epsilon \right) \left(1 - \frac{1}{k}\right)^2 + \text{GR-}\epsilon(k-3) \left(1 - \frac{1}{k}\right)^3 \dots \\
&\geq \frac{\text{OPT}(k)}{k} - \epsilon + \left(\frac{\text{OPT}(k)}{k} - \epsilon \right) \left(1 - \frac{1}{k}\right) + \left(\frac{\text{OPT}(k)}{k} - \epsilon \right) \left(1 - \frac{1}{k}\right)^2 \\
&\quad + \dots + \left(\frac{\text{OPT}(k)}{k} - \epsilon \right) \left(1 - \frac{1}{k}\right)^{k-2} + \text{GR-}\epsilon(1) \left(1 - \frac{1}{k}\right)^{k-1} \\
&\geq \left(\frac{\text{OPT}(k)}{k} - \epsilon \right) \sum_{i=0}^{k-1} \left(1 - \frac{1}{k}\right)^i \\
&= \text{OPT}(k) \left(1 - \left(1 - \frac{1}{k}\right)^k\right) - \epsilon k \left(1 - \left(1 - \frac{1}{k}\right)^k\right),
\end{aligned}$$

which we can bound from below by $(\text{OPT}(k) - \epsilon k) \left(1 - \frac{1}{e}\right)$ again by taking a limit $k \rightarrow \infty$. This shows

$$\begin{aligned}
\frac{\text{GR-}\epsilon(k)}{\text{OPT}(k)} &\geq \left(1 - \frac{\epsilon k}{\text{OPT}(k)}\right) \left(1 - \frac{1}{e}\right) = \left(1 - \frac{1}{e}\right) + \frac{1}{\text{OPT}(k)} \left(\frac{\epsilon k}{e} - \epsilon k\right) \iff \frac{\text{GR-}\epsilon(k) + \epsilon k \left(1 - \frac{1}{e}\right)}{\text{OPT}(k)} \geq 1 - \frac{1}{e}. \text{ Using} \\
\text{GR}(k) &\geq k \text{ (Proposition 12), we can reformulate } 1 - \frac{1}{e} \leq \frac{\text{GR-}\epsilon(k) + \epsilon k \left(1 - \frac{1}{e}\right)}{\text{OPT}(k)} \leq \frac{\text{GR-}\epsilon(k) + \text{GR-}\epsilon(k) \cdot \epsilon \left(1 - \frac{1}{e}\right)}{\text{OPT}(k)} = \\
&\frac{\text{GR-}\epsilon(k) \cdot \left(1 + \epsilon \left(1 - \frac{1}{e}\right)\right)}{\text{OPT}(k)}. \quad \square
\end{aligned}$$

ALGORITHM 5: A $\text{GREEDY-}\epsilon$ algorithm

Input: Numbers $n, m, k \in \mathbb{N}$ with $k \leq m$, set V of n voters, set C of m candidates.

Output: Committee $W \subseteq C$ of size k .

Let $W = \{\}$ be an empty set;

for $i = 1, \dots, k$, **do**

1. For all $c \notin W$, determine $\Delta(W, c)$.
2. Add to W any $c' \notin W$ for which $\Delta(W, c') \in [\max_{c \notin W} \Delta(W, c) - \epsilon, \max_{c \notin W} \Delta(W, c)]$, chosen uniformly at random.

end

Theorem 2. *Let an instance be non-trivial, $\delta > 0$, $\gamma \in (0, 1)$, and $k < t \leq m$. Then, w.p. at least $1 - \delta$, Algorithm 2 is $(1 - 1/e)\gamma$ -approximate for Max Cover with query complexity $O\left(\left(\frac{\gamma}{1-\gamma}\right)^2 km \log\left(\frac{km}{\delta}\right)\right) \in O_{\delta, \gamma, k}(m \log m)$.*

Proof. We show that the greedy query algorithm, in order to be $(1 - 1/e)\gamma$ -approximate for Max Cover with stated probability and number of queries, indeed needs the values of ϵ and ℓ as stated in Algorithm 2.

For $\gamma \in (0, 1)$, we take $\epsilon = \frac{(1-\gamma)e}{\gamma(e-1)}$ so that $\frac{\text{GR-}\epsilon(k)}{\text{OPT}(k)} \geq \gamma \cdot (1 - \frac{1}{e})$, using Lemma 13. We want to find ℓ so that with probability $1 - \delta$, all our estimations $\hat{\Delta}$ are within $\frac{1}{2}\epsilon$ of the true values Δ . For this, we use Hoeffding's inequality to bound the probability that a single estimate $\hat{\Delta}$ is more than $\frac{1}{2}\epsilon$ away from its corresponding true value Δ . Write $X_{W,c} : V \rightarrow \{0, 1\} : v_i \mapsto \mathbb{1}_{\{A(i) \cap (W \cup \{c\}) \neq \emptyset\}}(i) - \mathbb{1}_{\{A(i) \cap W \neq \emptyset\}}(i)$ for a random variable with expected value $\mathbb{E}[X_{W,c}] = \Delta(W, c)$. We can draw from $X_{W,c}$ by querying individual voters on $W \cup \{c\}$, then ℓ independent realizations with values in $\{0, 1\}$ yield sample mean $\hat{\Delta}(W, c)$, with $\mathbb{E}[\Delta(\hat{W}, c)] = \Delta(W, c)$. Hoeffding's inequality gives, for any estimate $\hat{\Delta}$ of Δ :

$$\mathbb{P}\left(|\hat{\Delta} - \Delta| \geq \frac{1}{2}\epsilon\right) \leq 2 \exp\left(-\frac{1}{2}\ell\epsilon^2\right). \quad (7)$$

There are k iterations and at most m estimates $\Delta(\hat{W}, c)$ per iteration so no more than $k \cdot m$ estimations in total. Thus, by independence, we have

$$\mathbb{P}\left(\forall i : |\hat{\Delta}_i - \Delta_i| < \frac{1}{2}\epsilon\right) \geq 1 - 2 \exp\left(-\frac{1}{2}\ell\epsilon^2\right) \cdot m \cdot k. \quad (8)$$

We want to find ℓ so that this value equals at least $1 - \delta$, and so we proceed to write

$$1 - 2 \exp\left(-\frac{1}{2}\ell\epsilon^2\right) \cdot m \cdot k \geq 1 - \delta \iff \exp\left(-\frac{1}{2}\ell\epsilon^2\right) \leq \frac{\delta}{2mk} \iff \ell \geq \frac{2}{\epsilon^2} \log\left(\frac{2mk}{\delta}\right). \quad (9)$$

This means, for any $\delta > 0$, any ℓ that is at least this large, will, with probability $1 - \delta$, produce (querying ℓ voters) estimates that are all at most $\frac{1}{2}\epsilon$ apart from the value that they estimate. Then we can invoke Lemma 13 to obtain the desired ratio. We take the smallest possible value: $\ell = \lceil \frac{2}{\epsilon^2} \log\left(\frac{2mk}{\delta}\right) \rceil$ and filling in for ϵ this results in

$$\ell = \left\lceil \frac{2}{\left(\frac{(1-\gamma)e}{\gamma(e-1)}\right)^2} \log\left(\frac{2mk}{\delta}\right) \right\rceil = \left\lceil \frac{2\gamma^2(e-1)^2}{(1-\gamma)^2 e^2} \log\left(\frac{2mk}{\delta}\right) \right\rceil \leq \left\lceil 0.8 \left(\frac{\gamma}{1-\gamma}\right)^2 \log\left(\frac{2mk}{\delta}\right) \right\rceil, \quad (10)$$

giving

$$k \cdot t \left\lceil \frac{m-k}{t-k} \right\rceil \cdot \ell \leq k \cdot t \left\lceil \frac{m-k}{t-k} \right\rceil \cdot \left\lceil 0.8 \left(\frac{\gamma}{1-\gamma}\right)^2 \log\left(\frac{2mk}{\delta}\right) \right\rceil \in O\left(\left(\frac{\gamma}{1-\gamma}\right)^2 km \log\left(\frac{km}{\delta}\right)\right) \quad (11)$$

queries to A in total. $\square$

B Proof of Theorem 6

We prove that LS- β -INCOMPLETE (Algorithm 4) can get arbitrarily close to the optimal approximation ratio for Max Cover using $O(m \log m)$ queries to voters. This result was stated as Theorem 6. First, we

discuss some changes we made to the complete local search algorithm of [11] (resulting in Algorithm 3), as these changes shine through in the incomplete local search algorithm.

The first change we made is that we have left out the greedy initialization. Although it speeds up the convergence time-wise, it would cost many extra querying rounds. It is not possible to query only for the initialization and use the gathered information for the local search phase, since candidates not elected in the initialization, or only elected in its last round, will not have query responses from all voters whereas that is necessary if the candidate would become part of the tentative solution at some point during the local search. However, it might be possible to save the information gathered in the initialization in a smart way. For now, we leave out this phase. For similar reasons, we omit the partial enumeration step that would yield the clean $1 - 1/e$ ratio. Secondly, we have adapted the stopping condition from $\frac{f((W \cup c') \setminus c)}{f(W)} \leq 1 + \beta n$ (the factor n is added because our scores range from $[0, 1]$ instead of $[0, n]$) to $f((W \cup c') \setminus c) - f(W) \leq \beta$. Before we can make any kind of statement about the performance of our local search querying algorithm, we need guarantees on the accuracy of our estimations $\hat{\Delta}$. We use Hoeffding's inequality to bound the probability that our estimates $\hat{\Delta}$ are far from the estimated true value Δ , but for that we need our samples to be bounded as well. For our stopping condition $\hat{\Delta}(W, c', c) = \hat{f}(W \cup \{c'\} \setminus \{c\}) - \hat{f}(W) < \beta - \epsilon$, these samples are really just any difference between two consecutive values of $(\alpha_i)_{i \in \mathbb{N}}$, since replacing c by c' , any voter can be represented one time less, the same number of times, or one time more. The samples are thus bounded from above by $\alpha_1 - \alpha_0 = \alpha_1$. However, since $\alpha_0 = 0$, samples for the stopping condition $\frac{f((W \cup c') \setminus c)}{f(W)} \leq 1 + \beta n$ are unbounded. Luckily, when $f((W \cup c') \setminus c) - f(W) \leq \beta$, then also $\frac{f((W \cup c') \setminus c)}{f(W)} \leq 1 + \frac{\beta}{f(W)}$, which is at most $1 + \beta n$ iff $f(W) \geq 1/n$. This will always be true as long as $\forall c_j \exists i : c_j \in A(i)$; all candidates have at least one approval, and $k \geq 3$, so that for any elected committee W , $f(W) \geq \alpha_3/n > 1/n$.

Theorem 6. *Let $|c_j| \geq 1 \forall j$, $\delta > 0$, $\gamma \in (0, 1)$, $m \geq t > k \geq 3$. Fix $\beta = C_2 \frac{1-\gamma}{\gamma k \log k}$ for some constant C_2 . Then, w.p. at least $1 - \delta$, Algorithm 4 is $(1 - 1/e)\gamma$ -approximate for Max Cover with query complexity $O\left(\left(\frac{\gamma k \log k}{1-\gamma}\right)^3 m \log\left(\frac{m \gamma k^2 \log k}{\delta(1-\gamma)}\right)\right) \in O_{\delta, \gamma, k}(m \log m)$.*

Proof. We show that the local search query algorithm, in order to be $(1 - 1/e)\gamma$ -approximate for Max Cover with stated probability and number of queries, indeed needs the values of ϵ and ℓ as in Algorithm 4. In the original approximation ratio (of Algorithm 3), we substitute γ for $(1 - 1/e)(1 - \gamma)$, so that the ratio becomes $(1 - 1/e) \cdot \gamma$ and we adapt the constant factor accordingly.

With $\beta = C_2 \frac{1-\gamma}{\gamma k \log k}$, the minimum step size of LOCAL SEARCH- β (Algorithm 3) is $C_2 \frac{1-\gamma}{\gamma k \log k}$, and every committee attains score at least 0 and at most α_k , so Algorithm 3 runs at most $\alpha_k \cdot \frac{\gamma k \log k}{C_2(1-\gamma)}$ iterations. We anticipate that querying will increase the amount of required iterations, because of the uncertainty inducing estimations. Depending on how many iterations we allow, values of ϵ (the estimation margin) and ℓ (the number of voters queried per round) will also differ. Thus, to make this dependency explicit, we add in a factor $\xi \geq 1$ and let Algorithm 4 have a total of at most $\xi \alpha_k \cdot \frac{\gamma k \log k}{C_2(1-\gamma)}$ iterations. We then compute the maximal value of ϵ this corresponds to.

Until the last run of the algorithm, we have $\hat{\Delta}(W, c', c) > \beta - \epsilon$, and when this is no longer true, the swap of c for c' is not made. We take ϵ so that all true improvements $\Delta(W, c', c)$ are at least size $\hat{\Delta}(W, c', c) - \epsilon > \beta - 2\epsilon$ and for that, we solve

$$\beta - 2\epsilon = C_2 \frac{1-\gamma}{\gamma k \log k} - 2\epsilon = C_2 \frac{1-\gamma}{\xi \cdot \gamma k \log k}, \quad (12)$$

for ϵ to obtain $\epsilon = \frac{\xi-1}{2\xi} \cdot C_2 \frac{1-\gamma}{\gamma k \log k}$, which is the maximum value of ϵ we may allow in order to have our original number of iterations (for Algorithm 3) multiplied by at most ξ .

We take ℓ such that with probability $1 - \delta$, all our estimations $\hat{\Delta}$ are within ϵ of the true values Δ . For this, we use Hoeffding's inequality to bound the probability that a single estimate $\hat{\Delta}$ is more than ϵ away from

its corresponding true value Δ . Write $X_{W,c',c} : V \rightarrow [-\alpha_1, \alpha_1] : v_i \mapsto \alpha_{h_i((W \cup \{c'\}) \setminus \{c\})} - \alpha_{h_i(W)}$ for a random variable with expected value $\mathbb{E}[X_{W,c',c}] = \Delta(W, c', c)$. We can draw from $X_{W,c',c}$, indirectly, by querying individual voters on $(W \cup \{c'\}) \setminus \{c\}$, then ℓ independent realizations yield sample mean $\hat{\Delta}(W, c', c)$, with $\mathbb{E}[\hat{\Delta}(W, c', c)] = \Delta(W, c', c)$. Regarding the range of $X_{W,c',c}$: For any committee W , any voter v_i will attain some score $\alpha_{h_i(W)} \in \{\alpha_0, \dots, \alpha_k\}$ and whenever a candidate $c \in W$ is replaced for another candidate $c' \notin W$, the voter can approve one less candidate than before, the same number of candidates as before, or one more candidate than before. A score α_i before a swap, can thus become any of scores $\{\alpha_{i-1}, \alpha_i, \alpha_{i+1}\}$ after. In Lemma 3, [11] prove that for all $i < k$, $\alpha_{i+1} > \alpha_i$ and $\alpha_{i+2} - \alpha_{i+1} \leq \alpha_{i+1} - \alpha_i$. Thus, $\{\alpha_i\}_i$ is an increasing sequence, but the increase declines monotonically. We may thus take $\alpha_1 - \alpha_0 = 1 - \frac{1}{e}$ for an upper bound on the difference between consecutive values in $\{\alpha_i\}_i$. Then we see that all our samples of $X_{W,c',c}$ take values in the interval $[-1 + \frac{1}{e}, 1 - \frac{1}{e}]$. Combining the above, Hoeffding's inequality gives, for any estimate $\hat{\Delta}$ of Δ :

$$\mathbb{P}(|\hat{\Delta} - \Delta| \geq \epsilon) \leq 2 \exp \left(-\frac{2\epsilon^2 \ell}{(2 - 2/e)^2} \right). \quad (13)$$

In each iteration we estimate $\Delta(W, c', c)$ based on all combinations (c', c) , so these are $(m - k) \cdot k$ estimations. This gives $(m - k) \cdot k \cdot \xi \alpha_k \cdot \frac{\gamma k \log k}{C_2(1-\gamma)}$ estimations in total over all iterations. The probability that for all i , the estimation $\hat{\Delta}_i$ of Δ_i is within ϵ of the true value, then becomes

$$\mathbb{P}(\forall i : |\hat{\Delta}_i - \Delta_i| < \epsilon) \geq 1 - 2 \exp \left(-\frac{2\epsilon^2 \ell}{(2 - 2/e)^2} \right) \cdot (m - k)k \cdot \xi \alpha_k \cdot \frac{\gamma k \log k}{C_2(1 - \gamma)}. \quad (14)$$

We can now choose ℓ such that this is at least $1 - \delta$:

$$\begin{aligned} 1 - 2 \exp \left(-\frac{2\epsilon^2 \ell}{(2 - 2/e)^2} \right) \cdot (m - k)k \cdot \xi \alpha_k \cdot \frac{\gamma k \log k}{C_2(1 - \gamma)} &\geq 1 - \delta \\ \iff 2 \exp \left(-\frac{2\epsilon^2 \ell}{(2 - 2/e)^2} \right) \cdot (m - k)k \cdot \xi \alpha_k \cdot \frac{\gamma k \log k}{C_2(1 - \gamma)} &\leq \delta \\ \iff \exp \left(-\frac{2\epsilon^2 \ell}{(2 - 2/e)^2} \right) &\leq \frac{1}{2} \delta \cdot \frac{1}{(m - k)k} \cdot \frac{1}{\xi \alpha_k} \cdot \frac{C_2(1 - \gamma)}{\gamma k \log k} \\ \iff -\ell &\leq \frac{(2 - 2/e)^2}{2\epsilon^2} \log \left(\frac{1}{2} \delta \cdot \frac{1}{(m - k)k} \cdot \frac{1}{\xi \alpha_k} \cdot \frac{C_2(1 - \gamma)}{\gamma k \log k} \right) \\ \iff \ell &\geq \frac{(2 - 2/e)^2}{2\epsilon^2} \log \left(\frac{2 \cdot (m - k)k \cdot \xi \alpha_k \cdot \gamma k \log k}{C_2(1 - \gamma)\delta} \right). \end{aligned} \quad (15)$$

This means, for any $\delta > 0$, any ℓ that is at least this size, will, with probability $1 - \delta$, produce (querying ℓ voters) estimates that are all at most ϵ apart from the value that they estimate. We take the smallest possible value of ℓ :

$$\ell = \left\lceil \frac{(2 - 2/e)^2}{2\epsilon^2} \log \left(\frac{2 \cdot (m - k)k \cdot \xi \alpha_k \cdot \gamma k \log k}{C_2(1 - \gamma)\delta} \right) \right\rceil.$$

The algorithm terminates when $\hat{\Delta}(W, c', c) < \frac{C_2(1-\gamma)}{\gamma k \log k} - \epsilon$, which implies $\Delta(W, c', c) < \frac{C_2(1-\gamma)}{\gamma k \log k}$. This is the original stopping condition of the algorithm without queries, so the original performance guarantees also hold here; the approximation ratio is $(1 - 1/e)\gamma$. Last we study the number of queries. In total there are

$$\begin{aligned} &\xi \alpha_k \cdot \frac{\gamma k \log k}{C_2(1 - \gamma)} \cdot t \left\lceil \frac{m - k}{t - k} \right\rceil \cdot \left\lceil \frac{(2 - 2/e)^2}{2\epsilon^2} \log \left(\frac{2 \cdot (m - k)k \cdot \xi \alpha_k \cdot \gamma k \log k}{C_2(1 - \gamma)\delta} \right) \right\rceil \\ &= \xi \alpha_k \cdot \left(\frac{\gamma k \log k}{C_2(1 - \gamma)} \right)^3 \cdot t \left\lceil \frac{m - k}{t - k} \right\rceil \cdot \left\lceil \frac{(2 - 2/e)^2}{2} \left(\frac{2\xi}{\xi - 1} \right)^2 \log \left(\frac{2 \cdot (m - k)k \cdot \xi \alpha_k \cdot \gamma k \log k}{C_2(1 - \gamma)\delta} \right) \right\rceil \end{aligned} \quad (16)$$

queries to A . Asymptotically in γ , δ , k and m (recall $k \ll m$), this is in $O \left(\left(\frac{\gamma k \log k}{1 - \gamma} \right)^3 m \log \left(\frac{m \gamma k^2 \log k}{\delta(1 - \gamma)} \right) \right)$. $\square$

C Lower bound for non-adaptive algorithms in the incomplete information setting

For the incomplete information setting, we have seen two algorithms. Both algorithms make use of information acquired during the run of the algorithm. Indeed, when selecting the query sets, we need these to contain the currently elected committee. We next present a lower bound for the query complexity of non-adaptive algorithms, which can not adapt their querying strategy to readily obtained information. The proof concerns the original setting, without matroid constraints, but the result extends to the problem over a matroid constraint, as it is a more general problem.

Theorem 14. *For any $k \geq 2$, $\epsilon > 0$, and $\gamma \in (\frac{1}{2}, 1)$, any non-adaptive committee selection algorithm that makes fewer than $\Omega(\frac{m^2 \epsilon}{1-\gamma})$ queries, is $(1 - 1/e)\gamma$ -approximate for Max Cover with probability at most ϵ .*

Proof. We denote a given non-adaptive committee selection algorithm by ALG. We define the class of instances that will serve as an example for the lower bound: let $k = 2p + r$, where $p \in \mathbb{N}$ and $r \in \{0, 1\}$, which means we can form any $k \in \mathbb{N}_{\geq 2}$ for the committee size. Partition the candidates into $C_1 \cup C_2 \cup \dots \cup C_p \cup D$, with $|C_i| = \lfloor (m-r)/p \rfloor$ for all $i \in \{1, \dots, p\}$, D containing the remaining candidates. Because $\sum_{i=1}^p |C_i| = p \cdot \lfloor (m-r)/p \rfloor \leq m-r$, we know $|D| \geq r$. For each i , we form $S_i = \{c_{i,1}, \dots, c_{i,\ell}\} \subseteq C_i$ by picking ℓ distinct candidates from C_i . The candidates are all chosen to join S_i with the same probability, independently of one another, and the assignment of these candidates to the numbers $1, \dots, \ell$ is also random. Moreover, this procedure happens independently for all C_i . We partition the voters into p parties $P_1, \dots, P_p$ and, if $r = 1$, one party R (if $r = 0$ there is no set R). Each party P_i contains $2/k$ of the voters and all voters in P_i approve only of candidates within S_i (so no one approves any candidate in $C_i \setminus S_i$). More specifically, for each P_i , we can again partition the candidates of C_i into two groups with non-overlapping approval sets: candidates $c_{i,1}, \dots, c_{i,\ell-1}$, are collectively approved by $1/2$ of the voters in P_i (so $1/(2k)$ voters in total) and $c_{i,\ell}$, that we call the *designated candidate*, is approved by the other half of P_i . The party R , if it exists, contains $1/k$ of the voters, all of whom approve precisely one candidate $d \in D$.

We say that ALG *covers* a set of candidates S if ALG ever submits a query set Q for which $|Q \cap S| \geq 2$. This means that, if for any of the parties P_i , ALG fails to cover S_i , all ℓ of these candidates are indistinguishable to ALG, given the definition of our query model. Indeed, all candidates within S_i are approved by the same number of voters, and we can only start to distinguish the structure of S_i if we combine $c_{i,\ell}$ and $c_{i,j}$ for some $j \in \{1, \dots, \ell-1\}$ in one query. Then, if a set S_i is not covered, the best ALG can do is pick candidates from S_i randomly, since without covering S_i it is still possible to distinguish which elements form the set S_i .

For each i , writing k_i for the number of candidates ALG selects from C_i , ALG picks the designated candidate with probability at most k_i/ℓ . Suppose that for all i , ALG picks some candidate $c_{i,j}$ for $j \in \{1, \dots, \ell-1\}$ and also picks the single approved candidate in R . We are allowed to assume this since this assumption facilitates ALG. This already gives a Max Cover score of $\frac{p+1}{2p+1}$. Then, in order to attain a $\gamma(1 - 1/e)$ -max cover score, ALG must still pick $\gamma(1 - 1/e)(2p+1) - (p+1)$ designated candidates. Then if $t = 1$, it is impossible for ALG to cover any of the S_i and we attain a $\gamma(1 - 1/e)$ -max cover ratio only if ALG is lucky enough to select at least $\lceil (1 - \frac{1}{e})\gamma + ((1 - \frac{1}{e})2\gamma - 1)p - 1 \rceil =: q$ of the designated candidates without having any information about which ones to pick. Clearly, for each C_i , the probability of picking the correct candidate is at most p/ℓ , as p is the total number of candidates we can still pick, and ℓ is the size of each set. The probability to pick at least $q \geq 1$ designated candidates is at most the probability to pick a single one: p/ℓ .

Now suppose that $t > 1$ and so we might be able to cover some S_i . Moreover, suppose that ALG knows the partition of candidates into $C_1 \cup \dots \cup C_p \cup D$ (so it will query within these sets) and knows the distribution of approvals within the sets C_i (but it doesn't know how this corresponds to the particular candidates). Moreover, we assume we can make at most $U \cdot m^2$ queries within each set C_i , for some constant U .

For any party P_i consider any set $S \subset C_i$ of size ℓ that is covered by a query set Q of size t . As soon as $\ell > t$ (so when m is large enough), we can split S into two parts: a part of size $2 \leq j \leq t$ that is *actually* covered by Q , and a part of size $\ell - j$ that is outside of Q . The number of sets $S \subset C_i$ of size ℓ that a single query can cover, then becomes

$$\sum_{j=2}^t \binom{t}{j} \binom{\lfloor (m-r)/p \rfloor - t}{\ell - j} \leq 2^t \binom{m/p}{\ell - 2} \leq 2^t \left(\frac{m}{p}\right)^{\ell-2} \quad (17)$$

if $\ell - 2 \leq \frac{m}{2p}$, because $\binom{n}{k}$ grows in n , and grows in k until $k = n/2$. Note that we require m to be large enough for this. With our Um^2 queries, at most

$$Um^2 \cdot 2^t \left(\frac{m}{p}\right)^{\ell-2} = Um^\ell 2^t p^{2-\ell} \quad (18)$$

sets of size ℓ within C_i can be covered in total. For m large enough, there are

$$\binom{\lfloor (m-r)/p \rfloor}{\ell} \geq \frac{(\lfloor (m-r)/p \rfloor - \ell)^\ell}{\ell!} \geq \frac{1}{\ell!} \left(\frac{m}{2p}\right)^\ell \quad (19)$$

sets of size ℓ in C_i in total, and since all sets were chosen to join S_i with the same probability, ALG covered S_i with probability at most

$$\frac{\ell! (2p)^\ell Um^\ell 2^t p^{2-\ell}}{m^\ell} = \ell! 2^{\ell+t} U p^2. \quad (20)$$

Suppose that ALG picked at least one of the candidates $c_{i,j}$ for $j \in \{1, \dots, \ell - 1\}$ for all i , then, like before, we need to pick at least q designated candidates in order to attain the desired ratio. ALG can do this by covering no S_i and picking all q designated candidates by chance, which happens, like proven above, with probability at most p/l . ALG can also cover one of the sets S_i and pick $q - 1$ designated candidates by chance, or any other combination is possible up to covering all sets S_i . Then, for $t > 1$,

$$\begin{aligned} \mathbb{P}(\text{ALG selects at least } q \text{ designated candidates}) &= \sum_{j=q}^p \mathbb{P}(\text{ALG selects } j \text{ designated candidates}) \quad (21) \\ &\leq \sum_{j=q}^p \sum_{h=0}^j \mathbb{P}(\{\text{ALG selects } j - h \text{ designated candidates by chance}\} \cap \{\text{ALG covers } h \text{ of the sets } S_i\}), \end{aligned}$$

where the inequality is due to the fact that covering a set S_i is necessary to distinguish its designated candidate, but might not be sufficient. For $h = 0$, we have bounded the probability above as p/ℓ . Each of the other at most $(p - q)p$ probabilities is bounded from above by the probability of covering one set S_i , which is at most $\ell! 2^{\ell+t} U p^2$. If we fix $U = \frac{\delta}{\ell! 2^{\ell+t} p^3 (p-q)}$ this becomes at most δ . The total probability is then at most $\frac{p}{\ell} + \delta$. Recall that we want this to be at most ϵ , which we can achieve, for instance, by fixing both terms to $\epsilon/2$. Since δ determines the query complexity, we can choose δ , so it suffices to show that $\frac{p}{\ell} \leq \frac{\epsilon}{2}$. Since $p = 2k - 1$, this condition is met when $\ell \geq \frac{k-1}{\epsilon}$.

As mentioned above, we have at most Um^2 queries and $U = \frac{\epsilon/2}{\ell! 2^{\ell+t} p^3 (p-q)}$, where $p = (k - 1)/2$. We want to denote the scaling of the complexity in terms of γ and ϵ . Since $p = (k - 1)/2$ and $q = \lceil (1 - \frac{1}{\epsilon})\gamma + ((1 - \frac{1}{\epsilon})2\gamma - 1)p - 1 \rceil$, we obtain

$$\frac{1}{p-q} \leq \frac{2}{1+k-2(1+((1-\frac{1}{\epsilon})\gamma-\frac{1}{2})k)} = \frac{2}{2k(1-(1-\frac{1}{\epsilon})\gamma)-1} \leq \frac{1}{k(1-\gamma)}, \quad (22)$$

where the last step required $2k\gamma/2 \geq 1/2$, which is true in case $k \geq 2, \gamma > 1/2$. It is not hard to see that $\frac{1}{\ell! 2^{\ell+t} p^3} \leq 1$, so that the final query complexity becomes $\frac{m^2 \epsilon}{1-\gamma}$. $\square$

Theorem 14 motivates the study of adaptive algorithms. It might be possible to extend this result to a larger exponent of m , as done in [14]), by adapting the instance. We leave this for future work.

D Inaccurate information: proofs of Theorem 7 and Theorem 8

We prove the upper and lower bound on achieving the optimal approximation ratio for Max Cover in the p -inaccurate query model. We stated these results as Theorem 7 and Theorem 8, respectively.

Recall that for accurate information, we write $A(i, j)$ to query the opinion of voter $v_i \in V$ on candidate $c_j \in C$, where the outcome is 1 in case of approval and 0 in case of disapproval. In the p -inaccurate query model, we instead query $A_p(i, j) = A(i, j) \oplus X$ where $X \sim \text{Bernoulli}(p)$, so that the outcome is flipped with probability p . This error is added independently for each pair (i, j) and each repeated sampling. For the upper bound, we give an algorithm which recovers the optimal approximation ratio with high probability. Our goal is to minimize the number of queries to $A_p(i, j)$.

ALGORITHM 6: GREEDY-INACCURATE

Input: Numbers $n, m, k \in \mathbb{N}$ with $k \leq m$, set V of n voters, set C of m candidates, parameters

$p \in (0, \frac{1}{2})$, $\delta > 0$.

Output: Committee $W \subseteq C$ of size k .

For all $v_i \in V$, $c_j \in C$, perform query $A_p(i, j)$ $U = \left\lceil 2 \frac{\log((nm)/\delta)}{1/\log(4p(1-p))} \right\rceil$ times and take the majority winner as the ‘true’ query response;

Let $W = \{\}$ be an empty set;

for $i = 1, \dots, k$, **do**

1. For all $c \notin W$, determine $\Delta(W, c)$.
2. Add to W any $c' \in \arg\max_{c \notin W} \Delta(W, c)$.

end

Theorem 7. Let $p \in (0, \frac{1}{2})$, $\delta > 0$, $n, m \in \mathbb{N}$. Then there exists an algorithm that is $(1 - 1/e)$ -approximate for Max Cover in the p -inaccurate query model w.p. at least $1 - \delta$ and with query complexity $O(nm \log(nm/\delta))$.

Proof. We adapt Algorithm 1, obtaining Algorithm 6. While Algorithm 1 simply queries $A(i, j)$ once for each of the nm voter-candidate pairs, Algorithm 6 instead performs each query $A_p(i, j)$ some $U > 1$ number of times and takes the majority winner to be the ‘true’ value, breaking ties arbitrarily. We now show that, taking U as in Algorithm 6, each of the nm majority votes are correct with probability at least $1 - \frac{\delta}{nm}$. Then, by independence, all together are correct with probability $1 - \delta$, in which case we obtain a $(1 - 1/e)$ -approximate solution. Each majority vote fails if a strict majority of the U samples is incorrect, and we can bound this failure probability using a Chernoff bound: Take $\Omega = \{0, 1\}^U$, $\mathcal{F} = \mathcal{P}(\Omega)$ and $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$ that satisfies $\mathbb{P}(\{\mathbf{v}\}) = (1-p)^y p^{U-y}$ when $\mathbf{v}$ contains y 1’s and $U-y$ 0’s. Accordingly, write $\mathbf{v}_{i,j}$ for a vector that is the outcome of U throws of (i, j) , where entry 0 means the outcome is corrupted and 1 means the outcome is correct. Then take $X : \Omega \rightarrow \mathbb{N} : \mathbf{v}_{i,j} \mapsto \sum_{l=1}^U v_{i,j,l}$ for the random variable that maps a vector to the number of correct outcomes and write $\bar{X} := U - X$ for the number of failures. A Chernoff bound then tells us that $\mathbb{P}(\bar{X} \geq \frac{1}{2}U) \leq \exp(-UD(\frac{1}{2}||p))$ [12] and we can reformulate to

$$\begin{aligned} \exp \left\{ -UD \left(\frac{1}{2} || p \right) \right\} &:= \exp \left\{ -U \left(\frac{1}{2} \log \left(\frac{1/2}{1-p} \right) + \frac{1}{2} \log \left(\frac{1/2}{p} \right) \right) \right\} \\ &= \exp \left\{ -U \frac{1}{2} \log \left(\frac{1/2}{1-p} \right) \right\} \exp \left\{ -U \frac{1}{2} \log \left(\frac{1/2}{p} \right) \right\} \\ &= \left(\frac{1}{4(1-p)p} \right)^{-\frac{1}{2}U}. \end{aligned} \tag{23}$$

We determine the value of U so that the above is at most $\frac{\delta}{nm}$:

$$\begin{aligned} \left(\frac{1}{4(1-p)p}\right)^{-\frac{1}{2}U} &\leq \frac{\delta}{nm} \iff -\frac{1}{2}U \leq \log_{\frac{1}{4p(1-p)}} \left(\frac{\delta}{nm}\right) \\ \iff U &\geq -2 \log_{\frac{1}{4p(1-p)}} \left(\frac{\delta}{nm}\right) = -2 \frac{\log \frac{\delta}{nm}}{\log \frac{1}{4p(1-p)}} = 2 \frac{\log \frac{nm}{\delta}}{\log \frac{1}{4p(1-p)}}. \end{aligned}$$

Fixing $\left\lceil 2 \frac{\log((nm)/\delta)}{1/\log(4p(1-p))} \right\rceil$ thus guarantees that all estimates are correct, and thus the approximation ratio is achieved with probability at least $1 - \delta$. $\square$

Although the above proof involved an adaptation of the Greedy algorithm, we note that we could have just as easily adapted the local search algorithm. Both algorithms will query all n voters on all m candidates. To deal with the p -inaccurate information setting, the above proof simply shows that repeating each of the nm queries $U = \log(nm/\delta)$ times guarantees that the nm queries are all correct with probability $1 - \delta$. This guarantee thus holds independently of what the algorithm does with the query outcomes, and also holds for the local search algorithm. This also implies that the above upper bound holds for the generalized problem of optimizing over a matroid constraint.

Theorem 8. *Let $p \in (0, \frac{1}{2})$, $\delta > 0$, $n, m \in \mathbb{N}$. Then any algorithm that is $(1 - 1/e)$ -approximate for Max Cover in the p -inaccurate query model w.p. at least $1 - \delta$ has expected query complexity $\Omega(nm \log(1/\delta))$.*

To establish the theorem, we adapt the proof of Theorem 9 in [21], using a result from multi-armed bandit theory [20] to lower bound the query complexity. The Kullback-Leibler divergence between two probability distributions is defined as $D(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$ for discrete probability functions P and Q , as long as, for all x , $Q(x) = 0 \implies P(x) = 0$ (otherwise we define it as $+\infty$). This reduces to $d(a, b) = a \log \frac{a}{b} + (1 - a) \log \frac{1-a}{1-b}$ for two Bernoulli distributions with parameters $a, b \in (0, 1)$.

Proof. Recall that we query voter $v_i \in V$ on candidate $c_j \in C$ using $A_p(i, j) = A(i, j) \oplus X$ where $X \sim \text{Bernoulli}(p)$. We will view each query $A_p(i, j)$ as an arm $a \in \mathcal{A}$ in a multi-armed bandit problem, and write $y_t := A_p(i, j)_t$ for the outcome of the t -th query. When $A(i, j) = 0$. Then $A_p(i, j) \sim \text{Bernoulli}(p)$. Conversely, when $A(i, j) = 1$ then $A_p(i, j) \sim \text{Bernoulli}(1 - p)$. We let P_a denote the Bernoulli distribution associated with action a : for any query index t and any $y \in \{0, 1\}$ we have $P_a(y) = \mathbb{P}(y_t = y | a_t = a)$. Let us refer to an arbitrary (possibly randomized) algorithm for the problem by ALG . We write $\mathbb{E}^\nu$ and $\mathbb{P}^\nu$ for the expectation and probability when the underlying instance is ν . Write L_ν for the set of optimal committees for instance ν . By assumption we have $\mathbb{P}^\nu(\text{ALG}(\nu) \in L_\nu) \geq 1 - \delta$ for any instance ν . We denote the random variable that portrays the number of queries taken when ALG terminates, by τ . We use the following lemma from multi-armed bandit theory, that is mentioned e.g. in [20], Chapter 15 (Lemma 15.1 and more specifically Exercise 15.7).

Lemma 15. *Let ν and ν' be any two bandit instances defined on the same set of arms $\mathcal{A}$, with corresponding observation distributions $\{P_a\}_{a \in \mathcal{A}}$ and $\{P'_a\}_{a \in \mathcal{A}}$. Let τ be the total number of queries made when the algorithm terminates, and let $\mathcal{E}$ be any probabilistic event that can be deduced from the resulting history $(a_1, y_1, \dots, a_\tau, y_\tau)$, possibly with additional randomness independent of that history. Then, for T_a the number of times action a is queried up to termination index τ , we have*

$$\sum_{a \in \mathcal{A}} \mathbb{E}^\nu[T_a] D(P_a || P'_a) \geq d(\mathbb{P}^\nu(\mathcal{E}), \mathbb{P}^{\nu'}(\mathcal{E})). \quad (24)$$

For any $k \geq 2$, consider an instance ν with m candidates and n voters. We partition the voters into m disjoint “parties” $P_1, \dots, P_m$. Voters in party P_i approve precisely c_i . Candidates $c_1, \dots, c_{m-k}$ (together C_1) all have one approving voter. These $m - k$ voters (together V_1) are alone in their party.

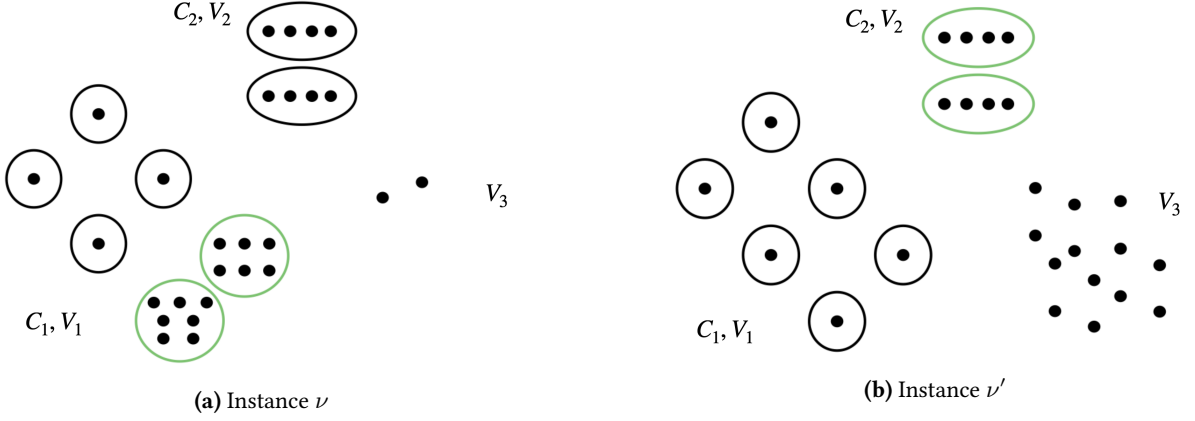


Figure 2: When $k = 2$, both instances have only one $(1 - 1/e)$ -approximate solution and these differ.

Candidates $c_{m-k+1}, \dots, c_m$ (together C_2) all have $u > \frac{1}{1-\frac{1}{e} \cdot k}$ approving voters (together V_2), so that the only $(1 - \frac{1}{e})$ -approximate solution consists of the k candidates in C_2 :

$$(k-1)u + 1 < \left(1 - \frac{1}{e}\right)ku \iff \left(\frac{1}{e}k - 1\right)u < -1 \iff u > \frac{1}{1 - \frac{k}{e}}. \quad (25)$$

This shows even holding in the committee $k-1$ candidates from C_2 and one candidate from C_1 , instead of k candidates from C_2 , yields an unforgivable loss in score. Finally we have V_3 , a set of voters that approve of no candidate (homeless voters). For any m, n , we can find constants $D, E \in (0, 1)$ such that $|C_1| \geq D \cdot m$ (which means we need $0 \leq m - Dm - k$) and $(1 - E)n - ku \geq |C_1|$ so that $|V_3| \geq E \cdot n$, so let's pick any such constants D and E . Now consider the instance ν' that is obtained by taking, $x := \lceil k(\frac{u}{1-\frac{1}{e}} - 1) \rceil$ homeless voters and distributing them (as) evenly (as possible) over k candidates $c_1, \dots, c_k \in C_1$. Other than that, all preferences remain unchanged. With this value of x , we have $x + k > \frac{ku}{1-\frac{1}{e}}$, which means that ν' has a unique $(1 - 1/e)$ -approximate solution in $c_1, \dots, c_k \subseteq C_1$, whereas the unique $(1 - 1/e)$ -approximate solution of ν was the set C_2 . Since both instances have only one optimal solution, and these differ, we have $L_\nu \cap L_{\nu'} = \emptyset$.

Now, regarding the inequality in Lemma 15, note the following. Firstly, $D(P_a || P'_a) = 0$ for all $a \notin \{(i, j) : i \in \{1, \dots, x\}, j \in \{1, \dots, k\}\}$. When $D(P_a, P'_a) \neq 0$, it equals $d(1 - p, p)$. Let $\mathcal{E}$ be the event that ALG outputs (the only element in) $L_{\nu'}$. Then, by assumption, $\mathbb{P}^\nu(\mathcal{E}) \leq \delta$ and $\mathbb{P}^{\nu'}(\mathcal{E}) \geq 1 - \delta$. Monotonicity of the discrete KL function d implies that $d(\mathbb{P}^\nu(\mathcal{E}), \mathbb{P}^{\nu'}(\mathcal{E})) \geq d(\delta, 1 - \delta)$.

So now we can rewrite the inequality in Lemma 15 as

$$\left(\sum_{i=1}^x \sum_{j=1}^k \mathbb{E}^\nu[T_{(i,j)}] \right) \cdot d(1 - p, p) \geq d(\delta, 1 - \delta), \quad (26)$$

or equivalently, $\left(\sum_{i=1}^x \sum_{j=1}^k \mathbb{E}^\nu[T_{(i,j)}] \right) \geq \frac{d(\delta, 1 - \delta)}{d(1 - p, p)}$. We can make such an adapted instance ν' for at least $\lfloor \frac{En}{x} \rfloor$ disjoint sets of x voters in V_4 (we gather these sets in a set we call $V_{4,x}$) and at least $\lfloor \frac{Dm}{k} \rfloor$ disjoint sets of k candidates in C_1 (we gather these sets in a set we call $C_{1,k}$):

$$\sum_{C_{1,k}} \sum_{V_{4,x}} \left(\sum_{i=1}^x \sum_{j=1}^k \mathbb{E}^\nu[T_{(i,j)}] \right) \geq \left\lfloor \frac{En}{x} \right\rfloor \left\lfloor \frac{Dm}{k} \right\rfloor \frac{d(\delta, 1 - \delta)}{d(1 - p, p)} \geq \left\lfloor \frac{E}{x} \right\rfloor \left\lfloor \frac{D}{k} \right\rfloor \cdot mn \cdot \frac{d(\delta, 1 - \delta)}{d(1 - p, p)}. \quad (27)$$

We rewrite the left hand side as

$$\sum_{C_{1,k}} \sum_{V_{3,x}} \left(\sum_{i=1}^x \sum_{j=1}^k \mathbb{E}^\nu[T_{(i,j)}] \right) \leq \sum_{v_i \in V} \sum_{c_j \in C} \mathbb{E}^\nu[T_{(i,j)}] = \mathbb{E}^\nu \left[\sum_{v_i \in V} \sum_{c_j \in C} T_{(i,j)} \right] = \mathbb{E}^\nu[\tau], \quad (28)$$

so we rephrase to $\mathbb{E}^\nu[\tau] \geq \lfloor \frac{E}{x} \rfloor \lfloor \frac{D}{k} \rfloor \cdot mn \cdot \frac{d(\delta, 1-\delta)}{d(1-p, p)}$. For the right hand side, note that p is a fixed constant in $(0, \frac{1}{2})$ and $d(\delta, 1-\delta) \in \Omega(\log(1/\delta))$ since $\delta \leq \frac{1}{2} - c$, hence, this instance ν serves to illustrate the lower bound. $\square$

E Discussion on the experiments

We discuss two aspects of the experiments. First, the pre-processing applied to the approval elections obtained from Polis, and second, how we determined the parameter ϕ that we used to instantiate the (q, ϕ) -resampling model of [32] to generate artificial approval elections.

Pre-processing of the data We did four pre-processing steps. First, of all 20 accessible Polis datasets, we remove the `london.youth.policing` dataset due to a low number of participants, and the `bg2025-volunteers` dataset because it contains far more statements than voters (all other sets have far fewer statements than voters, on average $m/n = 0.4$). Second, we removed statements with more than half of voters approving, as they made attaining a high Max Cover score trivial in some datasets (and are generally relatively weak statements). Third, we removed voters that didn't vote on any statement as well as those who didn't approve of any statements (as they are unsatisfiable).

Fourth and last, as mentioned in the main text already, we need to complete the data. On average across the 18 datasets, voters were presented with 18.2% of the statements. Their responses were 56.3% approvals, 27.3% disapprovals and 16.4% neutral votes. Since we require every query to be answered either as approval or disapproval we need to fill in unanswered queries and queries answered neutrally. We set all such votes to disapprove, which results in a worst-case for the Max Cover score of the algorithms.

Our motivation for this is twofold. First, we are not convinced the partial ballot is representative of the full ballot, making extrapolation a doubtful strategy. Indeed, in Polis, the query selection is random but weighed, taking into account, among other things, how divisive and recent a statement is, and how often it has answered neutrally. Moreover, often up to a third of voters only respond to a single query (often their own statement), while a small portion of voters vote on hundreds of statements. Second, since the majority of cast votes are approvals, extrapolation will make attaining a high Max Cover score trivial. Taking Max Cover as diversity measurement, as we do, this observation shows a diverse committee is practically achievable in the context of Polis (assuming the partial ballots are representative). But this also means that this setting is not of interest to study empirically. By setting all unknowns to disapprove, we obtain the worst-case Max Cover score across all extensions of the partial ballot (therefore, including the real extension), recovering an interesting empirical case-study: if we achieve good Max Cover score here, we also can in practice. However, note that although this is the worst-case for the Max Cover score of each individual algorithm, it is not necessarily worst-case with respect to the ratio between complete/accurate and incomplete/inaccurate algorithms. This would only be true in case the relative decrease in the incomplete/inaccurate Max Cover score is smaller than that of the complete/accurate Max Cover score.

Determining ϕ of the Polis data To determine ϕ , we compute the approvalwise vector [32, Definition 3] for each election, and then compute the approvalwise distance [32, Definition 4] between each election and the limit of the approvalwise vector [32, Section 5] of (q, ϕ) -resampling elections for $q = 0.0891$

and $\phi = 0, 1/100, 2/100, \dots, 1$. For each election, we determine the value of ϕ that minimizes this distance and then average over the 18 Polis datasets. This yields $\phi = 0.693$.