Artificial Intelligence 279 (2020) 103203

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Preference elicitation and robust winner determination for single- and multi-winner social choice ☆

Tyler Lu [*,1], Craig Boutilier [1]

Google Research, 1600 Amphitheatre Parkway, Mountain View, CA, USA

ARTICLE INFO

Article history:
Received 12 October 2014
Received in revised form 3 November 2019
Accepted 5 November 2019
Available online 12 November 2019

Keywords:
Social choice
Voting
Preference elicitation
Robust optimization
Partial preferences
Minimax regret
Decision theory

ABSTRACT

The use of voting schemes based on rankings of alternatives to solve social choice problems can often impose significant burden on voters, both in terms of communication and cognitive requirements. In this paper, we develop techniques for preference elicitation in voting settings (i.e., *vote elicitation*) that can alleviate this burden by minimizing the amount of preference information needed to find (approximately or exactly) optimal outcomes. We first describe robust optimization techniques for determining winning alternatives given partial preference information (i.e., partial rankings) using the notion of *minimax regret*. We show that the corresponding computational problem is tractable for some important voting rules, and intractable for others. We then use the solution to the minimax-regret optimization as the basis for vote elicitation schemes that determine appropriate preference queries for voters to quickly reduce potential regret. We apply these techniques to *multi-winner social choice problems* as well, in which a *slate* of alternatives must be selected, developing both exact and greedy robust optimization procedures. Empirical results on several data sets validate the effectiveness of our techniques.

1. Introduction

Effective schemes for the aggregation of user preferences are critical in settings where decisions must be made for a group of agents. Methods for preference aggregation have been widely studied under the guise of *social choice* within the realm of economics, political science, operations research and other areas, and have been gaining increasing attention within artificial intelligence. Indeed, the availability of data about the preferences of millions of individuals—afforded by recommender systems and product review sites, social media, e-commerce platforms that record consumer choice behavior, and related mechanisms—has made the practical solution to preference aggregation all the more pressing.

In social choice, *voting rules* of various sorts are commonly used for aggregating user preferences. While exceptions exist (e.g., simple methods like plurality and approval voting), most voting schemes make strong assumptions about the preferences provided by voters, in particular, assuming that each voter (or user) provides a complete preference ranking of all alternatives under consideration. While this requirement may be reasonable in some domains (e.g., high-stakes settings

☆ Parts of this paper appeared in: T. Lu and C. Boutilier, "Robust Approximation and Incremental Elicitation in Voting Protocols," *Proceedings of the Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*, pp. 287-293, Barcelona (2011) [63]; and "Multi-winner Social Choice with Incomplete Preferences," *Proceedings of the Twenty-third International Joint Conference on Artificial Intelligence (IJCAI-13)*, pp. 263–270, Beijing (2013) [65].

* Corresponding author.
E-mail addresses: tl@cs.toronto.edu (T. Lu), cebly@cs.toronto.edu (C. Boutilier).

[1] Research conducted while the author was at the Department of Computer Science, University of Toronto.

https://doi.org/10.1016/j.artint.2019.103203

such as political elections), social choice is increasingly being applied to lower-stakes, higher frequency domains, such as web search, product recommendation, meeting scheduling, and others. Insisting on complete preference information in such settings is often undesirable, since the number of options that need to be rated or ranked is often extremely large (and may be combinatorial in nature). As such, the cognitive and communication burden imposed by insisting on complete preference rankings may not be warranted given the stakes involved.

This suggests that making decisions with partial information about voter preferences may be desirable in many settings. First, it will often be the case that an optimal (group) decision, or implementation of the appropriate *social choice function*, can be realized with only partial preference information. Second, it may be acceptable to provide an *approximate* solution if it relieves the burden of preference specification, even when, in principle, full preferences could be determined or elicited. Finally, in certain situations, we may be unable to obtain complete preferences; e.g., in consumer choice settings, preference information is often limited to revealed choice data (e.g., which products have been purchased) or surveys, in which case complete preference information may simply be unavailable.

In this work, we address the problem of group decision making or preference aggregation with partial preferences. We consider both *single-winner* problems—in which a single consensus outcome or alternative is to be chosen for a group of users or voter population—and *multi-winner problems*—in which a set or *slate* of alternatives of some fixed size can be selected. While in general, the outcomes of many voting schemes cannot be determined without a large amount of information in the worst case [33,34], this does not diminish the practical imperative to minimize the amount of information elicited from voters in real-world settings.

We consider two key issues. First we consider the following problem: given *partial* information about voter preferences, how does one choose a suitable outcome or winning alternative?[2] Since we generally cannot guarantee that any alternative is the true winner given partial preferences, we propose the use of *maximum regret* to quantify the worst-case error of any selected alternative over possible realizations of voters' complete preferences. We then use *minimax regret* as our selection criterion, choosing as a winner the alternative that minimizes this error or maximum regret. This provides us with a form of *robust optimization*, and determines the alternative with the tightest quality guarantees given our uncertainty about true/complete voter preferences. We describe algorithms for computing minimax regret and robust outcome selection for several common voting rules (including Borda, Bucklin, maximin, and egalitarian[3]) in single-winner settings. We also describe exact and approximate algorithms for multi-winner settings, focusing on robust optimization for the Chamberlin-Courant scheme [26].

We emphasize that our framework deals with *informational approximation*. This stands in contrast to the large body of work on *computational approximation* of voting schemes given full preference information. Our approach shares much common ground with research on *possible and necessary winners* [56,96], as we discuss in depth below. Unlike this work, however, we propose a fairly general technique by which one actually chooses an outcome in settings with partial preferences.

The second issue we address is incremental *vote elicitation*. If the available voter preference information is limited, we will not generally be able to determine the true winning outcome; furthermore, the worst-case error associated with the robust winner (i.e., its max regret) may be larger than one is willing to allow. In such a case, additional preference information is required to improve decision quality. In this work, we show how to use our regret-based decision criterion to drive the process of *preference (or vote) elicitation*. Specifically, we use the solution to the robust optimization problem to choose *which queries to ask of which voters*. We develop a distribution-free heuristic called the *current solution strategy* that allows one to determine queries that quickly reduce minimax regret in practice. Experimental results on randomly-generated and real-world data sets—in both single- and multi-winner domains—show that high-quality, and in many cases optimal, outcomes can be determined even when voters express a relatively small proportion of their preferences.

The remainder of the paper is organized as follows. We review relevant background and related work in Section 2. In Section 3, we define minimax regret for single-winner voting models, and provide some theoretical observations of the concept. This includes demonstrating the intractability of computing minimax regret for several voting rules (e.g., Copeland and ranked pairs). In Section 4 we present polynomial time algorithms to compute the minimax regret for several common voting rules including positional scoring, maximin, Bucklin, and egalitarian voting rules. We discuss preference elicitation for single-winner problems in Section 5. We develop a meta-strategy called the *current solution strategy*—which uses the solutions provided by our robust optimization methods to determine which queries to ask—and describe its instantiation for two voting rules. We also demonstrate its effectiveness with computational experiments on several data sets. We extend our robust optimization approach to multi-winner problems in Section 6, providing both exact and greedy algorithms for the slate optimization problem, while in Section 7 we extend the current solution strategy to elicitation in multi-winner problems. We conclude in Section 8 with a discussion of possible extensions of this work.

2. Preliminaries and related work

In this section, we describe necessary background and required concepts, and provide a brief overview of related work. We begin by outlining the basic model of (single-winner) social choice used in this work. We then describe the form of

[2] In multi-winner settings, an outcome consists of a set of alternatives. We ignore this distinction in the remainder of this section, understanding that a single alternative must be replaced by a slate in multi-winner problems.

[3] We explain the choice of name for the egalitarian rule below.

partial preferences assumed, and continue by specifying the multi-winner social choice framework we analyze. We conclude with a discussion of related work on winner selection with partial information, vote elicitation, and the communication complexity of voting rules.

2.1. Preferences and voting rules

We begin with a review of basic concepts from social choice theory and describe several common voting schemes. In this section we focus on single-winner problems, deferring a discussion of multi-winner problems to Section 2.3. We refer the reader to [46,73] for further background on social choice and voting rules.

We assume a set of *agents*, or *voters*, $N = \{1, \ldots, n\}$ and a set of candidates or *alternatives* $A = \{a_1, \ldots, a_m\}$. Alternatives can represent any outcome space over which the voters have preferences (e.g., product configurations, restaurant dishes, candidates for a political office, public projects, etc.) and for which a single collective choice must be made. We refer to this problem in general terms as the *outcome selection problem*, and in the case of single-winner problems as *winner determination*.

We assume voter preferences for alternatives take the form of a total order or linear ranking of A. Let Γ be the set of *rankings* (or *votes*) over A (i.e., permutations over A), or equivalently, the set of bijections of the form $v : A \to \{1, \ldots, m\}$. Let $v_k \in \Gamma$ denote voter k's preference ranking. Let $v_k(a)$ denote the *rank* of a in v_k. We say *k prefers a_i* to a_j, written $a_i \succ_k a_j$ (or $a_i \succ a_j$ when k is clear from context), if $v_k(a_i) < v_k(a_j)$. We refer to a collection of votes $\mathbf{v} = \langle v_1, \ldots, v_n \rangle \in \Gamma^n$ as a *preference (or vote) profile*. Let V be the set of all such profiles.

Given a preference profile, we consider the problem of selecting a *consensus alternative*, requiring the design of a *social choice function* or *voting rule* $r : V \to A$ which selects a "winner" given voter rankings/votes.[4] Certain voting rules require a tie-breaking mechanism because more than one alternative may have the same optimum "score" (as we discuss below in the case of specific rules). In such cases we use $r(\mathbf{v})$ to represent both: (a) the winning alternative once the tie-breaking mechanism is applied; and (b) the set of *co-winners* generated prior to tie-breaking, in which case we write $a \in r(\mathbf{v})$ to indicate a is a co-winner. The usage will be clear from context.

We now outline several widely used and studied voting rules. *Plurality* is one of the most common rules: the alternative with the greatest number of "first place votes" wins (various tie-breaking schemes can be adopted). Plurality does not require that voters provide rankings; however, this "elicitation advantage" means that it fails to account for relative voter preferences for any alternative other than its top choice. The other schemes described below are more sensitive to relative preferences.

- *Positional scoring rules*: Assume a *positional scoring function (PSF)* α, where $\alpha(1), \ldots, \alpha(m)$ assigns a scalar value to each rank position, with $\alpha(i) \geq \alpha(i+1)$ for all $i \geq 1$. The *score* of a is $s_\alpha(a, \mathbf{v}) = \sum_k \alpha(v_k(a))$. The winner is the alternative with the greatest score. We write $\alpha_k(a) = \alpha(v_k(a))$ which can be interpreted as a measure of k's satisfaction with alternative a. The well-known *Borda count* is a positional rule with $\alpha(i) = m - i$. Plurality (with $\alpha(1) = 1$ and $\alpha(i) = 0$ for $i > 1$), k-approval (with $\alpha(i) = 1$ if $i \leq k$, $\alpha(i) = 0$ otherwise), k-veto ($\alpha(i) = 1$ if $i \leq m - k$, $\alpha(i) = 0$ otherwise) are also positional rules.
- *Egalitarian (maxmin fairness)*: Let $s_f(a, \mathbf{v}) = \min\{m - v_k(a) : k \in N\}$. The winner is the alternative with maximum score (i.e., the alternative whose worst rank position over all voters is the highest) [30].[5]
- *Copeland*: Define $W(a_i, a_j; \mathbf{v})$ to be 1 if more voters prefer a_i to a_j, 0.5 if tied, and 0 otherwise. Let $s_c(a_i) = \sum_{j \neq i} W(a_i, a_j; \mathbf{v})$. The alternative a with highest score $s_c(a)$ wins.
- *Maximin*: Let $N(a_i, a_j; \mathbf{v}) = |\{v_k : v_k(a_i) < v_k(a_j), k \in N\}|$ be the number of voters who prefer a_i to a_j. Let $s_m(a_i, \mathbf{v}) = \min_{j \neq i} N(a_i, a_j; \mathbf{v})$. The a with highest score $s_m(a, \mathbf{v})$ wins.
- *Bucklin*: The Bucklin score $s_B(a, \mathbf{v})$ is the smallest $k \in \{1, \ldots, m\}$ such that more than half of all voters rank a above position k. The winner is the alternative with smallest Bucklin score.[6]

We occasionally mention other voting rules below without defining them.

Notice that all of these voting schemes explicitly *score* alternatives w.r.t. voter preferences, implicitly defining some form of "societal utility," "degree of societal acceptance" or aggregate quality measure for each alternative. Indeed, this is true of many (though not all) voting schemes. In the sequel, we assume the existence of a *scoring function* $s(a, \mathbf{v})$ that measures the quality of any candidate given a preference profile $\mathbf{v}$. Specifically, consider some voting rule $r : V \to A$. We say a scoring function s is *consistent* with rule r iff $r(\mathbf{v}) \in \operatorname{argmax}_{a \in A} s(a, \mathbf{v})$ for all $\mathbf{v} \in V$. This is, of course a minimal requirement,

[4] *Consensus rankings* [3] could also be considered. The consensus ranking can be used for many purposes; e.g., the top-ranked alternative can be taken as the consensus winner, or we might select the top k alternatives in the consensus ranking in settings where multiple candidates can be chosen (say, parliamentary seats, or web search results [42]). Thus, consensus rankings can be used as a means of selecting winning sets.

[5] This voting rule seems not to have a well-established name. Congar and Merlin [30] call this rule "maximin," but this conflicts with the much-more established use of that name for the rule that follows in this list. Maxmin fairness is a natural name, but also not well-established. The term *egalitarian social welfare* is used commonly in the social choice literature (see, e.g., [6,28,83]) and reflects precisely the criterion adopted in this rule. Given that we adopt a "social welfare" perspective later in this paper, we adopt the term *egalitarian voting rule* in our work. We note that the rule bears some connection to the *threshold aggregation* scheme [2], but the latter actually is a lexicographic (leximin) criterion.

[6] This is sometimes known as "simplified Bucklin". In another variation, the alternative with the highest k-approval score wins, where k is the simplified Bucklin score.

since any voting rule can be defined using an indicator function as the score. However, all rules discussed above have "natural" scoring functions. Our approach to robust optimization will exploit this fact. When there are ties among the highest scoring candidates, usually some form of tie-breaking is used. None of our results are tied critically to any specific form of tie-breaking.

2.2. Partial preferences

One obstacle to the widespread use of voting schemes based on full preference rankings is the informational and cognitive burden imposed on voters, and concomitant ballot complexity. This partly explains the popularity of plurality voting in many jurisdictions [73], and the fact that much advocacy of more expressive methods (e.g., approval voting [20]) often falls short of proposing the use of full rankings. Elicitation of sufficient, but *partial* information about voter rankings could alleviate some of these concerns by reducing this burden. At the same time, as discussed above, there are settings where we might be unable to access complete vote profiles (e.g., when using consumer choice data), or when the stakes involved do not warrant the cost of eliciting full rankings.

For these reasons, we consider below winner determination with partial preferences. We assume that the *partial ranking* of a voter k takes a very general form, namely, a partially ordered set over domain A, or equivalently (the transitive closure of) a consistent collection of *pairwise comparisons* of the form $a_i \succ a_j$.[7] Most natural constraints on preferences, including the responses to natural queries (e.g., pairwise comparison, positional, top-t, and other queries) can be represented in this way.[8]

Let p_k be the partial ranking of voter k. A *completion* of p_k is any vote v_k that extends p_k. Let $C(p_k)$ denote the set of *completions* of p_k, that is, the set of all (complete) votes v_k that extend p_k. We introduce the following notation:

- $Nec_k(a \succ b)$ iff $a \succ b$ in all completions of p_k; i.e., a is necessarily preferred to b by k.
- $Pos_k(a \succ b)$ iff $a \succ b$ in some completion of p_k; i.e., a is possibly preferred to b by k.
- $Inc_k(a, b)$ iff $Pos_k(a \succ b)$ and $Pos_k(b \succ a)$; i.e., neither a nor b is known to be preferred to the other by k, so they are "incomparable" given the knowledge of k's preferences encoded by p_k.

A *partial profile* is a collection of partial votes $\mathbf{p} = \langle p_1, \ldots, p_n \rangle$. Let $C(\mathbf{p}) = C(p_1) \times \ldots \times C(p_n)$ be the set of *completions* of $\mathbf{p}$.

For time-complexity analysis of the algorithms in this paper, we assume that the representation of a voter k's partial preference ranking is such that assessment (e.g., look up) and addition of any pairwise preference $a \succ_k b$ takes $O(1)$ time. This enhances readability and is reasonable in many practical situations where the number of alternatives m is a small constant. Partial preferences may be represented using set membership data structures. For any partial preference p, we maintain two set membership data structures, `ancestor` and `descendant` where $a \in \texttt{ancestor}[b]$ if $a \succ b$ and $a \in \texttt{descendant}[b]$ if $b \succ a$. Adding a pairwise comparison to p may result in additional implied comparisons via transitive closure, hence an addition operation can be called multiple times. Set membership data structures can be implemented in a variety of ways, for example, using red-black trees where look up queries take $O(\log m)$ time.

2.3. Multi-winner models

While considerable research in computational social choice has focused on single-winner choice problems, only recently has much attention been paid to the algorithmic aspects of *multi-winner choice problems* [13,62,67,71,77,78,87]. Multi-winner problems are of critical importance in many settings. The canonical example is that of electing a legislature or committees using proportional representation [26,72]. However, multi-winner models are also appropriate for resource allocation, product recommendation, multiple facility location problems, consumer segmentation and a variety of other problems where either: (a) a limited number of options can be offered; (b) budget constraints preclude personalized recommendations; or (c) budget can be traded off against group satisfaction [55,62].

Multi-winner problems deal with the selection of a set or *slate of alternatives* $\bar{a} \subseteq A$. A voter's satisfaction with a slate is a function of their satisfaction with the options in the selected slate, and constraints are generally placed on the set of *feasible slates*. In this work, we focus on one of the conceptually simplest such multi-winner models: (i) we assume that voter satisfaction with candidates is given by some scoring function (e.g., the Borda PSF) induced by their ranking, and that a voter's satisfaction with the slate is given by the score of their most preferred alternative on the slate; and (ii) feasible slates are those with size $|\bar{a}| \leq K$. In other words, we can select up to K candidates and each voter derives benefit from their most preferred candidate.

More formally, given a preference profile $\mathbf{v}$, we define the *score* $S(\bar{a}, \mathbf{v})$ of a K-set $\bar{a} \subseteq A$ and the *optimal K-set* $\bar{a}^*_{\mathbf{v}}$ as follows:

---

[7] A set of comparisons is consistent if its transitive closure is anti-reflexive. Unless otherwise stated, we assume all sets of pairwise comparisons describing a single (partial) vote are consistent.

[8] One exception involves constraints that are naturally disjunctive; e.g., a response to the question "What alternative is ranked t^{th}?" cannot be mapped to a set of pairwise preferences unless the positions t are queried in ascending or descending order.

$$S(\bar{a}, \mathbf{v}) = \sum_{i \in N} S_i(\bar{a}) \quad \text{where} \quad S_i(\bar{a}) = \max_{a \in \bar{a}} \alpha_i(a), \tag{1}$$

$$\bar{a}^*_{\mathbf{v}} = \operatorname*{argmax}_{|\bar{a}| \leq K} S(\bar{a}, \mathbf{v}). \tag{2}$$

(We suppress dependence of S on α since the PSF will be fixed and clear from context.) Note that we use the term K-set to refer to sets of size *at most* K rather than exactly K for brevity. In most settings below, sets of size less than K may be optimal and might be preferred for tie-breaking, though an optimal set of size K always exists.

When α is the Borda PSF, this corresponds to the Chamberlin-Courant rule [26].[9] This slate optimization problem can be viewed as a *segmentation problem* [55]; and it is also a special case of *budgeted social choice*, specifically, the *limited choice* form of the problem [62]. More general forms of the Chamberlin-Courant rule [72] and budgeted social choice [62] allow for *assignment functions* that map voters to specific options (e.g., to ensure balanced representation, or budget feasibility); but here we assume that the only constraint is on the number of options selected. Sets of size less than K offer no advantage over those of size K in this case.

2.4. Related work

In this section, we provide a brief overview of work in computational social choice, specifically in voting settings, that addresses issues of partial information and preference elicitation. In some cases, we defer a deeper discussion to later sections, where we draw explicit comparisons to our own approach.

Voting and social choice. We have provided a brief discussion of various voting rules above, specifically focused on some of the rules we examine in our work. There are many excellent overviews of social choice and voting to which we refer the interested reader for more background, including general overviews [46,73] as well as material with a more specific computational focus [21,27,32].

Multi-winner social choice. Below we focus on preference elicitation and robust optimization for multi-winner problems that correspond to the Chamberlin-Courant [26] scheme as defined above. An important variant proposed by Monroe [72] requires the use of an assignment function—that associates each voter with a specific winner—so that candidates each represent an equal number of voters.

These problems can be viewed as instances of *segmentation problems* [55], in which one must solve an optimization problem for a number of different "customers," each of whom has a different objective function, hence different values for a given solution. Full personalization corresponds to selecting an optimal solution for each customer given her objective (or preferences), but this is often not feasible due to budget, capacity or other constraints on the collective solution. Partial personalization can be realized by *segmenting* customers into groups, and determining the best solution for each group. Kleinberg et al. [55] consider problems in which at most K solutions can be selected from a set, corresponding to the multiple winner problem we consider (if customer objectives are given by positional satisfaction), as well as problems in which K is variable, and each solution incurs a fixed cost γ (and the aim is to find the solution with greatest net return).

Lu and Boutilier [62] generalize this problem somewhat further, developing a *budgeted social choice* model, of which the specialization corresponding to the simple multi-winner problem with positional satisfaction—which they dub *limited choice*—is also considered. They show that the limited choice problem is NP-hard, but present a greedy algorithm that has a $1 - \frac{1}{e}$ approximation ratio (and performs well in practice). Skowron et al. [88] provide a stronger PTAS for this problem. Procaccia et al. [78] prove NP-hardness of a variant of the Chamberlin-Courant scheme, while Potthoff and Brams [77] provide an integer programming formulation of Chamberlin-Courant (limited choice).

Skowron et al. [88] address the complexity of approximate winner determination for both the Chamberlin-Courant and Monroe schemes (analyzing satisfaction, dissatisfaction and egalitarian variants). They show hardness of approximation in some cases, and provide approximation schemes (including the PTAS mentioned above) for others. Skowron et al. [87] analyze some of these algorithms empirically. Oren and Lucier [67] analyze the regret of online variants of the budgeted social choice problem.

Facility location [48] and maximum coverage problems [29] all bear close connection to these problems as well. For instance, Skowron and Faliszewski [85] draw a connection between maximum coverage with "bounded frequencies" and an approval variant of Chamberlin-Courant, and derive approximation schemes for several versions of this problem. We are unaware of any work that considers either robust optimization of slates in multi-winner problems in the presence of an incomplete preference profile, nor any work that considers the incremental elicitation of voter preferences in such a setting.

Of course, a variety of other multi-winner voting rules and social choice problems have been considered in the literature (e.g., [5,7,24,43,44,86]), more than can be surveyed here.

[9] Note that Chamberlin-Courant [26] specify their model in terms of *misrepresentation* and attempt to minimize total degree of misrepresentation rather than maximize total satisfaction; but the notions are equivalent for determining an optimal slate.

Voting and partial information. Determining the winners of elections under partial information is a problem that has received significant attention. Among the most important concepts when dealing with partial information are *possible and necessary winners* [56]. We say a is a *possible winner* under $\mathbf{p}$ if and only if there is some $\mathbf{v} \in C(\mathbf{p})$ such that $r(\mathbf{v}) = a$; and a is a *necessary winner* under $\mathbf{p}$ iff $r(\mathbf{v}) = a$ for all $\mathbf{v} \in C(\mathbf{p})$. Computationally, the possible winner question is often quite difficult (NP-complete), while necessary winner computation can be easy (polynomial time) or difficult (co-NP-complete) depending on the rule [56,96]. Considerable additional work has addressed the complexity of possible and necessary winner determination, in both unweighted and weighted voting (e.g., [10,12,33,58,93]).

Possible and necessary winners are purely epistemic notions—they do not prescribe a method for actually *choosing* winners in general under partial profiles (a point to which we return below). If one has some prior distribution over voter preferences, winners can be estimated probabilistically given a partial vote profile. Bachrach et al. [8] consider a (restricted) probabilistic analog of the possible winner problem, and show that counting the number of completions in $C(\mathbf{p})$ of partial profile $\mathbf{p}$ is #P-hard, but provide a polynomial time (randomized) approximation algorithm for the problem. Hazon et al. [49] consider a model that allows for arbitrary (i.i.d.) preference distributions and develop a dynamic programming method to compute the probability that a candidate a wins under several different voting rules.

Preference elicitation and communication. The *elicitation* question has been studied from a theoretical perspective, addressing whether winners for some voting rules can be determined with partial voter preferences (rankings). Unfortunately, worst-case results are generally discouraging. Conitzer and Sandholm [34] demonstrate that the communication complexity of several common voting protocols, such as Borda and Copeland, is $\Theta(nm \log m)$, essentially requiring communication of full voter preferences in the worst-case. Indeed, determining which votes to elicit to determine a winner is NP-hard in many schemes (e.g., Borda) [33,94]. We refer the reader to Boutilier and Rosenschein [18] for a survey of the information and communication requirements of various voting rules.

Despite the theoretical complexity of partial elicitation, practical means of eliciting partial rankings and making decisions with partial preferences are vital. This is precisely the problem we address in this work. Kalech et al. [53] were among the first to consider practical vote elicitation schemes for several different score-based voting rules (e.g., Borda, range voting). They propose two elicitation schemes using specific query types, and adopt possible and necessary winners as their primary solution concept. Their *iterative voting* method determines a true winner (with no approximation) and proceeds in *rounds*.[10] At each round, voters are queried for their next best candidate (and by round t have answered the equivalent of a *top-t query*, as defined below). At the end of each round, necessary and possible winners are computed with respect to the current partial vote profile: if every possible winner is a necessary (co-)winner, the process stops and returns the set of necessary winners. (The algorithm could also be terminated once *any* necessary winner is found.) On small random and real-world vote profiles (up to 30 voters and 50 alternatives), this scheme can reduce the number of alternatives ranked by voters by up to 10–40%, with the larger gains possible when user preferences are more uniform.

Kalech et al. [53] also propose *greedy voting*, which proceeds for a fixed number of rounds t. Given the current partial profile $\mathbf{p}^t$ at round t, the minimal and maximal possible scores $s(a, r)$, over all completions r of $\mathbf{p}^t$, of each candidate a are computed. Then each voter ranks the set of q alternatives with the largest minimum scores (maximum scores are used to break ties), for some small q. Since termination must occur after t rounds, necessary winners may not result, so possible winners are returned. This scheme cannot guarantee winner optimality, nor any bounds on quality. Kalech et al. show empirically that the (post hoc) quality of the resulting winners in small domains is reasonably high and, of course, improves with the number of rounds permitted. We contrast these methods with our techniques below.

Subsequent work on elicitation has focused on several rather distinct approaches. These include the use of approximation (to select alternatives that are "close" to winning—as we do in this work—or selecting the true winner with high probability). Some of these methods exploit probabilistic models of voter preferences, while others are distribution free, and are sometimes designed with specific voting rules in mind. The use of machine learning methods to predict missing preferences is also commonplace. We discuss such approaches here.

Lu and Boutilier [64] propose the use of sampled vote profiles—from historical data or from some known or learned probability distribution over voter preferences—to determine the value of k required so that top-k voting yields an (approximate) winners with high probability, providing PAC-based sample complexity results on the number of profiles needed. Oren et al. [45,74] consider the use of top-k voting to determine winners with high probability, for positional scoring rules (and Copeland). They provide lower and upper bounds on the k required for various distributional models of preferences (e.g., impartial culture and Mallows) and scoring rules. They also consider "zero-elicitation" protocols (effectively, $k = 0$).[11]

Lee et al. [60] shows how winner approximation with high probability allows one to circumvent general lower bounds on exact winner determination for both Borda and the Condorcet winner. Specifically, only a logarithmic (in the number of alternatives and the approximation terms) number of pairwise comparisons are needed, which is independent of the number of voters. (Note this work makes no assumptions about preference distributions.) In a related vein, Dey and Bhattacharyya [36] consider sampling the (complete) rankings of a subset of voters and derive upper and lower bounds on the sample

[10] This is distinct from iterative voting schemes in which voters change their votes in response to the current vote profile, which go by the same name (e.g., [61,70,82].)

[11] Baumeister et al. [9] analyze top-k voting schemes, but from the perspective of campaign design and possible winner determination.

complexity for a large number of voting rules required to ensure an approximate winner is discovered with high probability. In terms of strategyproof elicitation, Lee [59] describes a technique to efficiently elicit ε-approximate winners that applies to all tournament voting rules that is ε-strategyproof and also preserves a form of voter privacy.

Azari Soufiani et al. [90] exploit a probabilistic model of voter utilities given attributes of both voters and candidates, and apply a Bayesian experiment design approach to elicit full rankings. However, their scheme is not directly driven by a specific decision criterion such as minimax regret, but rather emphasizes maximum likelihood estimation (MLE). Caragiannis et al. [25] also consider the design of voting rules under distributional assumptions (e.g., Mallows) for the purposes of maximizing likelihood with respect to some underlying true ranking.

An alternative way of exploiting probabilistic information is to use it as the basis for predicting missing preferences. Doucette et al. [40] use classification to predict missing preferences in voter ballots and run voting rules on these *imputed* ballots. They provide no theoretical guarantees for this approach (though do analyze the impact of classifier performance), but analyze the method experimentally on a variety of data sets.

Other work on vote elicitation considers several tasks related to determining the minimal information required to select a winner, in both general cases, and under assumptions of the structure of voter preferences. Ding and Lin [39] analyze the computation of a "deciding set" of queries for a particular candidate—these are queries where any consistent set of responses will fully determine whether a candidate must be a winner or loser. However, this cannot be construed as an interactive elicitation scheme. Dery et al. [35] develop practical elicitation heuristics for range voting [89] that exploit a probabilistic model of voter preferences, and consider both entropy reduction and alternatives with highest posterior winning probability to generate queries. This probabilistic model is updated in a manner akin to collaborative filtering in recommender systems and is shown to offer significant reductions in queries needed in several practical group recommendation tasks.

Zhao et al. [97] develop techniques for group elicitation under randomized voting (including Borda and plurality) assuming a Plackett-Luce probabilistic model of user preferences, and consider a fairly broad class of queries—specifically, which k of a set of l alternatives are most preferred, which includes pairwise comparisons as a special case—as well as alternatives characterized by a set of features.

Conitzer [31] was among the first to demonstrate that structural assumptions on preferences can make elicitation easier than the general case: he shows that single-peaked preferences (under certain knowledge assumptions) can be elicited with a linear number of comparison queries per agent (and this serves as a lower bound on the number of queries needed for aggregate ranking). Dey and Misra [37,38] analyze additional structural assumptions on voter preferences (respectively, single-crossing preferences and single-peaked preferences on trees for various tree parameters) and provide upper and lower bounds on query complexity for a several restricted elicitation modalities. Another form of structure on preferences is related to the multi-attribute (or feature) structure of the alternatives themselves. The work of Zhao et al. [97] mentioned above uses feature structure; Benabbou et al. [11] extend the minimax-regret approach we develop here to multi-attribute domains using linear programming. We note that minimax regret has been used for multi-attribute optimization and preference elicitation in a variety of single-agent (non-aggregate) settings, as discussed below.

Other uses of minimax regret for preference elicitation. Finally, we note that *minimax regret*—the robustness criterion we adapt to the problem of winner determination under partial profiles and to drive our elicitation process—has been used rather widely for the same purposes in single-agent decision making and preference/utility elicitation. This includes work on multi-attribute optimization problems [15–17,19,22,23,84,95], combinatorial optimization of item slates [92], and Markov decision processes [79,80]. It has also been used in (multi-agent) mechanism design to minimize the amount of utility function information revealed to direct mechanisms such as the VCG mechanism [50–52]. Boutilier [14] provides an overview of such techniques. We also note that minimax regret has been used as a robustness criterion for optimization problems (e.g., linear programs) in which there is data or objective function uncertainty [1,4,57].

3. Robust winner determination: the single-winner case

In this section, we consider single-winner voting problems, and address the question of *robust winner determination* given partial information about voter preferences. We introduce the notion of *minimax regret* as a robustness criterion for such problems in Section 3.1. We discuss the relationship of minimax regret with the notions of possible and necessary winners in Section 3.2, and use this relationship to show that computing minimax regret is computationally intractable for certain voting rules (e.g., Copeland, ranked pairs). In the following section, Section 4, we investigate the computation of minimax regret, developing polynomial time algorithms for various voting rules, including Borda, maximin, Bucklin and egalitarian, based on the construction of worst-case completions of partial profiles.

3.1. Minimax regret

Let r be a voting rule, defined using some natural scoring function $s(a, \mathbf{v})$ such that $r(\mathbf{v}) \in \operatorname{argmax}_{a\in A} s(a, \mathbf{v})$ as described above.

Suppose we have a partial profile $\mathbf{p}$ and we are forced to make a decision in the face of this incomplete information. We distinguish our *partial information* setting from the question of aggregating preferences of voters whose preferences reflect genuine *incomparability* (see, e.g., [75])—as mentioned above, we view two alternatives as incomparable *with respect to the* *information encoded in a voter's partial preference* if it does not allow one to conclude that either is preferred to the other. Unfortunately, the notions of necessary and possible winners do not resolve this issue satisfactorily: necessary winners are not guaranteed to even exist given arbitrary partial profiles; and possible winners can only be used to narrow the set of options rather than to prescribe an actual winner. Here we propose the use of the *minimax regret* solution concept. This concept has been used for robust decision making, and for driving preference elicitation, in a variety of single-agent domains [16,19,23] and in mechanism design [51]; but our work is the first application of the notion to voting and (rank-based) social choice.

Intuitively, we measure the quality of any proposed winner a given $\mathbf{p}$ by considering how far from optimal a could be in the worst case (i.e., given any completion of $\mathbf{p}$). The minimax optimal solution is any alternative that is nearest to optimal in the worst case. More formally, we define:

$$Regret(a,\mathbf{v}) = \max_{a'\in A} s(a',\mathbf{v}) - s(a,\mathbf{v}) \tag{3}$$

$$= s(r(\mathbf{v}),\mathbf{v}) - s(a,\mathbf{v})$$

$$PMR(a,a',\mathbf{p}) = \max_{\mathbf{v}\in C(\mathbf{p})} s(a',\mathbf{v}) - s(a,\mathbf{v}) \tag{4}$$

$$MR(a,\mathbf{p}) = \max_{\mathbf{v}\in C(\mathbf{p})} Regret(a,\mathbf{v})$$

$$= \max_{a'\in A} PMR(a,a',\mathbf{p}) \tag{5}$$

$$MMR(\mathbf{p}) = \min_{a\in A} MR(a,\mathbf{p}) \tag{6}$$

$$a^*_{\mathbf{p}} \in \operatorname{argmin}_{a\in A} MR(a,\mathbf{p}) \tag{7}$$

$Regret(a,\mathbf{v})$ is the loss (or regret) of selecting a as a winner, given true vote profile $\mathbf{v}$, instead of choosing the optimal alternative under rule r (equivalently, under scoring function s).[12] $PMR(a,a',\mathbf{p})$ denotes the *pairwise max regret* of a relative to a' given partial profile $\mathbf{p}$. This is simply the worst-case loss—over all possible complete realizations, $C(\mathbf{p})$, of a partial profile—of selecting alternative a rather than a'. Notice that pairwise max regret can be negative. $MR(a,\mathbf{p})$ is the maximum regret (or max regret) of a, in other words, the worst-case loss associated with selecting a rather that selecting a true (score-maximizing) winner. We can view this as *adversarial selection* of a complete profile $\mathbf{v}$ to maximize the loss between our chosen alternative a and the true winner under $\mathbf{v}$. Our aim is to choose the alternative a with *minimum max regret*: $MMR(\mathbf{p})$ denotes minimax regret under partial profile $\mathbf{p}$, while $a^*_{\mathbf{p}}$ denotes the minimax optimal alternative.[13] This gives us a form of robustness in the face of vote uncertainty: every alternative has worst-case error at least as great as that of $a^*_{\mathbf{p}}$.

3.2. Relationship to possible and necessary winners

All voting rules discussed above have natural scoring functions (this fact is exploited in the algorithms for necessary winner determination developed by Xia and Conitzer [96]). The notions of possible and necessary winners can be generalized to allow for ties when scores are used: a *necessary co-winner* is a candidate with maximum score in all possible completions of a partial profile (hence, even if the tie-breaking rule used by r goes against it, a is still "as good" a candidate as any winner actually selected by r); possible co-winners are defined similarly [96].

Notice that if $MMR(\mathbf{p}) = 0$, then the minimax winner $a^*_{\mathbf{p}}$ has the same score or utility as the winner in any completion $\mathbf{v} \in C(\mathbf{p})$; i.e., $a^*_{\mathbf{p}}$ is guaranteed to be optimal. While this does not imply there is a necessary winner under $\mathbf{p}$ (due to tie-breaking), $MMR(\mathbf{p}) = 0$ iff there is a necessary co-winner. Thus for any rule r we have, by setting $\varepsilon = 0$:

Observation 1. The max regret decision problem for any voting rule r (i.e., does alternative a have $MR(a,\mathbf{p}) \le \varepsilon$) is at least as computationally hard as the necessary co-winner problem for r.

This implies that computing minimax regret is co-NP-hard for the following voting rules [96]: Copeland, ranked pairs, (balanced or unbalanced) voting trees, and plurality with runoff. This observation does *not* imply, however, the easiness of either max regret or minimax regret computation when the necessary co-winner problem is easy; but we describe polynomial time algorithms to compute minimax regret for several important voting rules in Section 4.

The relationship between max/minimax regret and possible winners is more complicated. For certain scoring rules the minimax winner $a^*_{\mathbf{p}}$ must be a possible winner under $\mathbf{p}$. For example, in plurality voting, if we assume each partial vote is either empty or lists only the top alternative, the minimax winner is the alternative a with the greatest number of votes (which are by assumption "first-place" votes). This a is also a possible winner since we can complete all empty votes by placing a at the top. However, in general, we have:

[12] See Smith [89] who uses score-based regret to measure the performance of various voting rules, including range voting.
[13] We informally write as if the optimal candidate is unique, but there can be several alternatives a that minimize max regret.

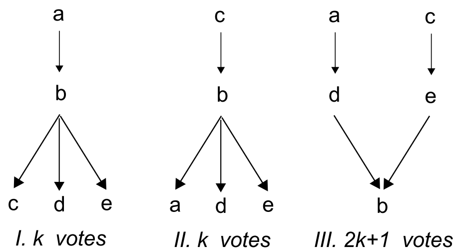

Fig. 1. A partial profile $\mathbf{p}$ where the minimax alternative is not a possible winner (under 2-approval). Alternative b has score $2k$ in every completion. Either a or c must be at the top of every vote in set III, so one of them must receive at least $k+1$ approval points from set III. Hence $\max(s(a), s(c)) \geq 2k+1$, and a and c are both possible winners, while b is not. Now, $MR(b, \mathbf{p}) = k+1$ (a completion that puts a at the top of all votes in set III would give a a score of $3k+1$, the maximum possible). But $MR(a, \mathbf{p}) = 2k+1$: if we select a, the adversary will place c and e above a in each vote in set III, setting $s(a) = k$ and $s(c) = 3k+1$. $MR(c, \mathbf{p}) = 2k+1$ by similar reasoning.

Observation 2. There are voting rules and partial preference profiles for which the regret-minimizing alternative is not a possible winner.

Fig. 1 illustrates this observation, showing a vote profile in which, under the 2-approval voting rule, the alternative that minimizes max regret is not a possible winner.[14] Both possible winners in this example have a poor 2-approval score under *some* completion of the votes, while a compromise candidate that cannot win under *any* completion has a much higher guaranteed score (i.e., lower max regret) than either possible winner. This suggests that using the notion of possible winners to select winners with partial votes can be problematic for some voting rules. Indeed, there would appear to be no general way to ensure a possible winner isn't *far from being optimal* without using max regret to quantify this risk. The fact that the minimax winner $a^*_{\mathbf{p}}$ is not a possible winner is not problematic in our view, but if one insists on selecting from the set of possible winners, max regret could at least be used to aid in that selection (i.e., to choose the possible winner with least max regret). Still we take max regret to be the more fundamental notion for winner determination with partial information.

4. Computing single-winner minimax regret

Minimax regret can often be solved as a mixed integer program (MIP) [16,19] or a search problem [23] in a variety of decision problems. In our voting context, a straightforward MIP formulation—with $O(|N||A|)$ integer variables to capture rank placement in each vote completion—would be prohibitively expensive to solve (likely beyond the reach of general MIP solvers for all but relatively small problems, especially for the types of high-frequency, low-stakes domains described in the introduction). However, for certain voting rules and preference constraints, we can greatly simplify minimax regret computation by directly considering properties of the worse-case completions of voter profiles without directly computing them. We illustrate this for several voting rules in this section. Our constructions are tightly related to those used by Xia and Conitzer [96] to demonstrate polynomial time algorithms for necessary winners for the positional scoring, maximin, and Bucklin rules. Indeed, their constructions can be viewed as attempting to maximize the difference in score between a proposed winner and an "adversarially chosen" alternative. We adapt these ideas to minimax regret, and extend the analysis to egalitarian voting.

4.1. Exploiting pairwise max regret

To demonstrate the polynomial time computability of minimax regret for specific voting rules below, we explicitly compute the pairwise max regret $PMR(a, w, \mathbf{p})$ of all $\frac{m(m-1)}{2}$ ordered pairs of alternatives (a, w) such that $a \neq w$, where a is a *proposed winner* and w is an *adversarial witness*. With PMR in hand, we can readily determine minimax regret using Eqs. (5) and (6). We thus need only show that PMR can be computed in polynomial time.

A scoring function is *(additively) decomposable* if $s(a, \mathbf{v}) = \sum_i s(a, v_i)$; i.e., if it is the sum of vote-wise scores. This implies that (pairwise) regret is decomposable, since

$$Regret(a, w, \mathbf{v}) = s(w, \mathbf{v}) - s(a, \mathbf{v}) \tag{8}$$

$$= \sum_i s(w, v_i) - \sum_i s(a, v_i) \tag{9}$$

$$= \sum_i [s(w, v_i) - s(a, v_i)]. \tag{10}$$

[14] In 2-approval, the top two candidates in each voter ranking each receive one point, and the winner is that with highest total approval score.

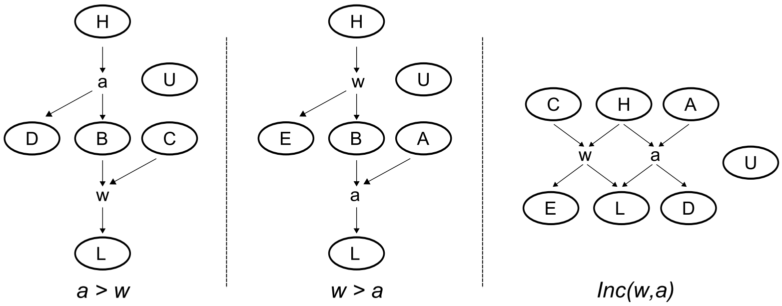

Fig. 2. An illustration of the three possible relations between alternative a and adversarial alternative or "witness" w in any partial vote p. Other alternatives stand in a specific relation to a and w, with each oval representing one of the possible relationships (see text). To maximize a partial vote's contribution to pairwise max regret $PMR(a, w, \mathbf{p})$, linearizations of p require placing the groups of alternatives indicated by the ovals in specific positions relative to a and w in a way that depends on the scoring function.

Given a set of partial votes p_i, their completions by an adversary can be undertaken independently, so we can compute *PMR* by independently choosing the completions v_i of each p_i that maximize v_i's local regret:

$$PMR(a, w, \mathbf{p}) = \max_{\mathbf{v} \in C(\mathbf{p})} s(w, \mathbf{v}) - s(a, \mathbf{v}) \tag{11}$$

$$= \sum_i \max_{v_i \in C(p_i)} s(w, v_i) - s(a, v_i). \tag{12}$$

All positional scoring functions are decomposable in this way.

4.2. Positional scoring rules

We illustrate our constructions by first examining the relatively simple case of computing $PMR(a, w, \mathbf{p})$ for a *linear* positional scoring rule, effectively, a positional rule that uses positive affine transformation of the Borda score.[15] Since *PMR* is decomposable, we determine, for any partial vote p, the completion v with maximum contribution to *PMR*.

Fig. 2 illustrates the three different cases we need to consider in any partial vote p: (1) p implies $Nec(a \succ w)$; (2) p implies $Nec(w \succ a)$; and (3) p implies $Inc(w, a)$. In each case, the remaining alternatives stand in one of several possible relationships with a and w, as indicated by the ovals, each representing the set of alternatives that fall into the following categories:

- H: those alternatives that are (known to be) preferred to both a and w in the partial vote p.
- B: those alternatives that are between a and w in p. If $Nec(a \succ w)$ then these are preferred to w, while a is preferred to them. If $Nec(w \succ a)$ then these are preferred to a, while w is preferred to them. If $Inc(w, a)$, then there are no such alternatives.
- L: those alternatives that are dispreferred to both a and w.
- D: those alternatives that are dispreferred to a but incomparable to w. If $Nec(w \succ a)$, then there are no such alternatives.
- E: those alternatives that are dispreferred to w but incomparable to a. If $Nec(a \succ w)$, then there are no such alternatives.
- C: those alternatives that are preferred to w but incomparable to a. If $Nec(w \succ a)$, then there are no such alternatives.
- A: those alternatives that are preferred to a but incomparable to w. If $Nec(a \succ w)$, then there are no such alternatives.
- U: those alternatives that are incomparable to both a and w.

The lack of preference arrows between certain sets (ovals) does not mean that pairwise preferences are not known between elements of some of these pairs. For instance, the preference $c \succ u$ for some $c \in C$ and $u \in U$ may be part of the partial vote. This relation does not play a role in constructing a completion for *PMR* below, so it is ignored. Of course, not all such pairwise preferences across distinct sets are viable—for instance, it is not possible to have $u \succ c$, since this would imply $u \succ w$, contradicting the definition of U. Similarly, elements within these sets may have known preferences within p, for instance, we may know $c \succ c'$ for some $c, c' \in C$. Indeed, we exploit these "internal" known preferences in determining queries during preference elicitation (see Section 5.1). We use these categories in analyzing other voting rules below as well.

Returning to linear positional scoring rules, in the first case, we have $Nec(a \succ w)$, so p's contribution to *PMR* must be negative. It is easy to see that pairwise regret is maximized with a completion v that minimizes the positional gap between

[15] Linear means that an alternative's score is a linear function of its rank in v, hence the *difference* in two rank positions uniquely determines their difference in score. Such rules are Borda-like for all practical purposes.

a and w (i.e., maximizing the adversary's (negative) advantage). To minimize the gap, it suffices to: (i) order set D below w, i.e., assume $w \succ d$ for all $d \in D$; (ii) order set C above a, i.e., assume $c \succ a$ for all $a \in A$; and (iii) order elements of set U either above a or below w. These orderings can be arbitrary (one need only maintain consistency within and across the sets in question). This implies that the (negative) contribution to *PMR* is exactly $-(|B|+1)$. Note that we needn't compute an actual linearization of p, but simply need to determine the cardinality of the set B.

The second case of $Nec(w \succ a)$ proceeds similarly (see figure), but instead we *maximize* the positional gap between w and a (i.e., maximize the advantage of w over a). This is accomplished by ordering sets E, A and U (arbitrarily) between w and a. Hence, the contribution to *PMR* by p in the second case is $|B \cup F \cup E \cup U| + 1 = m - |H \cup L| - 1$. Finally, in the third case of $Inc(a, w)$, the (positive) advantage of w over a is maximized by ordering w over a and placing sets E, A and U between the two.

Computing *PMR* thus requires, for each partial vote, categorizing all alternatives as belonging to the relevant sets described above as indicated in Fig. 2. This is a simple matter: one compares each alternative a' to both a and w in the partial vote (assuming the transitive closure of p is given), classifying it into the appropriate set, which takes $O(m)$ time. This implies that $PMR(a, w, \mathbf{p})$ is computable in $O(nm)$ time for linear scoring rules. With $O(m^2)$ pairs, computing $MMR(\mathbf{p})$ (and the optimal $a^*_{\mathbf{p}}$ and its witness) takes $O(nm^3)$ time. In many practical settings, m can be treated as a small constant relative to n, in which case our algorithms scale linearly in n.

With linear positional scoring rules, arbitrary placement of alternatives that do not influence the positional gap between w and a (e.g., set U when $Nec(a \succ w)$) is allowed. For nonlinear rules, the size of the gap *and the position* of both a and w can influence w's advantage when $Nec(a \succ w)$. However, the required placement can be found by simply examining splits of the set U of different cardinalities to determine how many to place above a and below w to minimize a's advantage over w—again, this can be accomplished in $O(m)$ time.

More formally for non-linear positional scores, the adversarial completion of a partial vote p when $Nec(a \succ w)$ requires that items from U are placed either above a or below w, since placing these items between a and w results in smaller regret. It only matters the number of items k (and not the identity of those items) from U that is placed above a or below w to maximize regret—because the regret depends only on the rank positions of a and w. Since there are $|U|$ items, we can check how many items $k \in \{0, \dots, |U|\}$ from U should be placed above a with the remainder $|U| - k$ items placed below w to maximize regret. For each k, we can determine the rank positions of a and w and hence their pairwise regret. As there are at most m items in U, there is an additional $O(m)$ computational cost relative to linear positional scoring rules (where the choice of k does not change the regret value).

Certain special cases can be treated more efficiently; e.g., if a positional rule has score gaps that are *monotonically non-increasing* (i.e., $s_i - s_{i+1} \geq s_{i+1} - s_{i+2}$, satisfying a discrete form of "convexity") then U is placed above a (and if non-decreasing, below w). In any case, minimax regret computation remains $O(nm^3)$. These observations show that:

Theorem 3. *Minimax regret can be computed in $O(nm^3)$ time for any positional scoring rule.*

4.3. Maximin voting

We now consider pairwise max regret computation for the maximin voting rule, a non-decomposable voting rule that requires more intricate computation. However, it is "semi-decomposable" in the sense that we can compute independent completions of each partial vote p_i, for each of a number of alternatives, and then aggregate the resulting scores.

To compute *PMR*, we recall the definition of the maximin scoring function, and embed it within the definition of *PMR*:

$$\begin{aligned} PMR(a, w, \mathbf{p}) &= \max_{\mathbf{v} \in C(\mathbf{p})} s_m(w, \mathbf{v}) - s_m(a, \mathbf{v}) \\ &= \max_{\mathbf{v} \in C(\mathbf{p})} \left[s_m(w, \mathbf{v}) - \min_{a' \neq a} N(a, a'; \mathbf{v}) \right] \\ &= \max_{\mathbf{v} \in C(\mathbf{p})} \left[s_m(w, \mathbf{v}) + \max_{a' \neq a} (-N(a, a'; \mathbf{v})) \right] \\ &= \max_{\mathbf{v} \in C(\mathbf{p})} \max_{a' \neq a} \left[s_m(w, \mathbf{v}) - N(a, a'; \mathbf{v}) \right] \\ &= \max_{a' \neq a} \left[\max_{\mathbf{v} \in C(\mathbf{p})} s_m(w, \mathbf{v}) - N(a, a'; \mathbf{v}) \right]. \end{aligned} \tag{13}$$

Let $M_{a'} = \max_{\mathbf{v} \in C(\mathbf{p})} s_m(w, \mathbf{v}) - N(a, a'; \mathbf{v})$ denote the quantity inside the square brackets of Eq. (13). This represents the worst-case difference (over completions of $\mathbf{p}$) of the maximin score of w and the number of votes, $N(a, a'; \mathbf{v})$, in which a is preferred to a'.

We proceed by describing an algorithm for computing the pairwise max regret $PMR(a, w, \mathbf{p})$, which can then be used for computing the minimax optimal alternative. Given fixed a and w, we must consider all alternatives $a' \neq a$ in order to compute the expression in Eq. (13). We fix such an a', and consider worst-case completions of each p_i for this a'.

- If $Nec_i(w \succ a)$ or $Inc_i(a, w)$, we maximize the advantage of w over a by placing w as high as possible and a as low as possible in the worst-case completion of p_i, using the techniques described in Section 4.2.
- If $Nec_i(a \succ w)$, we consider all other alternatives b ($b \neq a$, $b \neq w$):
  - if $Nec_i(a \succ b)$ and $Inc_i(b, w)$, we place b below w;
  - if $Nec_i(b \succ w)$ and $Inc_i(b, a)$ we place b above a;
  - if $Inc_i(b, a)$ and $Inc_i(b, w)$, we: (1) place b below w if $b \neq a'$; or (2) place b above a otherwise;
  - otherwise, we place b arbitrarily, subject to the partial preference constraints.

Let $\mathbf{v}'$ be the completion of the partial votes p_i in $\mathbf{p}$ as specified above. As above, specific completions need not be constructed, as the computations below can be largely performed using the cardinality of "sets" of alternatives relating w, a and a'. We compute $M_{a'} = s_m(w, \mathbf{v}') - N(a, a'; \mathbf{v}')$ for each a', and take the largest such $M_{a'}$ to be the pairwise max regret.

We now argue that this algorithm computes *PMR* correctly. Assume that we have a worst-case completion $\mathbf{v}^* \in C(\mathbf{p})$ that is a maximizer of $M_{a'}$ (i.e., $s_m(w, \mathbf{v}^*) - N(a, a'; \mathbf{v}^*) = M_{a'}$). We claim that we can reconfigure $\mathbf{v}^*$ into another completion $\mathbf{v}'$ of $\mathbf{p}$ (exactly as described above) such that $s_m(w, \mathbf{v}') - N(a, a'; \mathbf{v}') = M_{a'}$. For any v_i^* and its corresponding partial vote p_i, the reconfiguration is the same as for positional scoring rules in the cases when $Nec_i(w \succ a)$ or $Inc_i(a, w)$—that is, we place w as high as possible and a as low as possible in the resulting ranking.

The case $Nec_i(a \succ w)$ is different. For an alternative b such that $Nec_i(a \succ b)$ and $Inc_i(b, w)$, we must place b below w. This does not change $N(a, a'; \{v_i^*\})$, and does not decrease the maximin score of w, even if $b = a'$. If b is such that $Nec_i(b \succ w)$ and $Inc_i(b, a)$, then we place b above a: this does not change the maximin score of w and will not increase $N(a, a'; \{v_i^*\})$, even if $b = a'$.

Finally, consider alternatives b where $Inc_i(b, a)$ and $Inc_i(b, w)$ (note that for all other cases, b must be placed above a, between a and w, or below w). If $b \neq a'$, then we place b below w—this will not change $N(a, a'; \{v_i^*\})$ and does not decrease the maximin score of w. If $b = a'$ and a' is not above a in v_i^*, then we move it above a, as this decreases $N(a, a'; \{v_i^*\})$ by 1 and decreases maximin score of w by at most 1, hence their difference can only increase. In all cases above, we can reconfigure votes v_i^* into $\mathbf{v}'$ such that $N(a, a'; \mathbf{v}^*) \geq N(a, a'; \mathbf{v}')$ and $s_m(w, \mathbf{v}^*) \leq s_m(w, \mathbf{v}')$, hence $s_m(w, \mathbf{v}') - N(a, a'; \mathbf{v}') \geq s_m(w, \mathbf{v}^*) - N(a, a'; \mathbf{v}^*) = M_{a'}$. Of course, the construction $\mathbf{v}'$ does not require using $\mathbf{v}^*$ as a starting point—this only serves to prove the optimality of $\mathbf{v}'$ as constructed by the algorithm above.

By this method, we see that the relevant completion for a' given $\mathbf{p}$ can be computed in $O(nm)$ time. By Eq. (13), pairwise max regret can be computed in time $O(nm^2)$, by computing $M_{a'}$ for all $a' \neq a$ and choosing the largest such value. Consequently, computing minimax regret using this algorithm takes $O(nm^4)$ time:

Theorem 4. *Minimax regret can be computed in $O(nm^4)$ time for the maximin voting rule.*

4.4. Bucklin voting

As with maximin, the Bucklin score of a candidate is not decomposable across voters, so it too requires a more intricate approach to constructing profile completions. Since the standard definition of the Bucklin score involves selection of the alternative with the smallest Bucklin score, we work with the inverted score $b(a, \mathbf{v}) = m - s_b(a, \mathbf{v})$ to remain consistent with our "score maximizing" definitions.

As above, we wish to compute $PMR(a, w, \mathbf{p})$ with respect to (inverted) Bucklin score $b(\cdot, \mathbf{v})$. To compute *PMR*, we solve the following decision problem, for various values of t: is $PMR(a, w, \mathbf{p}) > t$. The ability to solve this problem efficiently means we can compute *PMR* in polynomial time by solving this problem for $t = m - 2, m - 3, \ldots, 0$, in this order. $PMR(a, w, \mathbf{p})$ is then the largest t^* for which $PMR(a, w, \mathbf{p}) > t^* - 1$. (Note that for Bucklin, *PMR* must be one of $\{0, \ldots, m - 1\}$).

This decision problem can be further broken down by solving a sequence of sub-problems which asks whether there is a completion $\mathbf{v} \in C(\mathbf{p})$ such that $s_b(a, \mathbf{v}) > k$ and $s_b(w, \mathbf{v}) \leq k'$ for fixed values $k, k' \in \{0, \ldots, m - 1\}$. If such a completion exists, this implies $PMR(a, w, \mathbf{p}) > k - k'$. Hence, to solve the *PMR* decision problem above for any fixed t, we can solve the subproblem for all values $k \leq m - 1$, with its corresponding $k' = t - k$.

To solve the sub-problem, we consider the relevant worst-case completions of each partial vote, based on different cases, as we did for positional scoring and maximin voting. If either $Nec_i(w \succ a)$ or $Inc_i(a, w)$ holds, then the worst case completion places w as high as possible and a as low as possible in i's vote, as in the case of positional scoring rules.

If $Nec_i(a \succ w)$ holds, then more subtle reasoning is required. First, it is clear that all alternatives in the set D (see Fig. 2, $a \succ w$) must be placed below w, and those in set C above a. This, of course, only serves to improve w's Bucklin score and possibly worsen a's score, which is our aim. At issue, however, is how many alternatives from the set U should be placed above a and how many below w, since we have a choice (obviously none should be placed between a and w). Consider the following two conditions:

- Condition 1: a is ranked below the kth position.
- Condition 2: w is ranked at or above position k'.

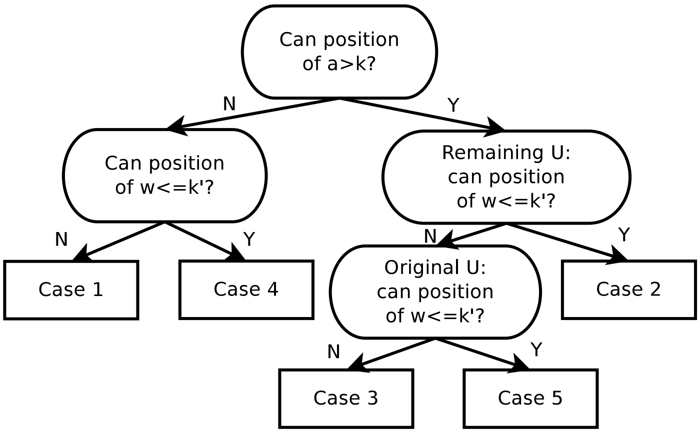

Fig. 3. Classifying p_i, with $Nec_i(a \succ w)$, into one of the five cases, used in answering the sub-problem of whether there exists a completion $\mathbf{v}$ such that $s_B(a, \mathbf{v}) > k$ and $s_B(w, \mathbf{v}) \leq k'$. The decision tree asks whether alternatives in U can be arranged to be placed above a or below w so as to achieve the desired rank positions. See Fig. 2 for definition of U.

We consider five distinct cases (in all cases, we assume that all alternatives apart from those in U are now positioned appropriately):

1. If p_i cannot be arranged such that either Condition 1 or 2 is satisfied (i.e., there are not enough alternatives in U to push a below position k or to push w to or above position k'), then we complete p_i in arbitrary fashion.
2. If p_i can be arranged to simultaneously satisfy both conditions, we use that completion.
3. If p_i can be arranged to satisfy Condition 1 but not Condition 2, we use that completion.
4. If p_i can be arranged to satisfy Condition 2 but not Condition 1, we use that completion.
5. Otherwise p_i can be arranged to satisfy either of Conditions 1 or 2, but not both simultaneously. We tentatively "skip" these partial votes and complete them as the last step of the algorithm.

To solve the subproblem, we use two counters: c_a keeps track of the number of partial votes that fall into Cases 1–4 above in which a's rank is below k; and c_w counts the number of partial votes that fall into Cases 1–4 above in which w's rank is at or above k'. Notice that partial votes within these first four cases offer no scope to trade off the relative position of a and w to satisfy our two conditions. Once these are counted, we need to construct suitable completions of the remaining partial votes, all of which fall into Case 5. Specifically, to validate the subproblem, we need $\lfloor n/2 \rfloor + 1 - c_a$ of the Case 5 votes to be completed in such a way that a's rank is below k, and $\lfloor n/2 \rfloor + 1 - c_w$ votes to be completed so that w's rank is at or above k'. Note that these two requirements are exclusive: the completion of a Case 5 partial vote 5 can only satisfy one of these two requirements.

Now if the number of partial votes n_s satisfying Case 5 is such that $n_s < 2\lfloor n/2 \rfloor + 2 - c_a - c_w$, then we cannot construct the required number of partial votes to satisfy the Bucklin score requirements, therefore we conclude that answer to the sub-problem is false. Otherwise we can satisfy the requirements by configuring $\lfloor n/2 \rfloor + 1 - c_a$ of these n_s votes so that a's position falls below k (by placing alternatives from U above a), and configuring $\lfloor n/2 \rfloor + 1 - c_w$ of the votes so that w's position is at least in rank k' or better (by placing U below w).

Testing the five cases to which a particular p_i belongs can be done in $O(m)$ time (see Fig. 3): we first check if U is large enough to place sufficiently many alternatives above a to make its rank position greater than k. If this is not possible, we check whether positioning U below w pushes w to position k' or above. If this can be achieved, the partial vote falls in case 4, if not it falls in case 1. If a can be pushed below position k, then we check whether the remaining alternatives in U (i.e., those left after removing just enough alternatives to push down a's position) can be placed below w to make its position at least k'. If so, then it falls into case 2. If not, we check whether the original set U can be placed below w to meet its positional requirement. If so, it falls into case 5, and otherwise it is case 3. Therefore the running time to solve the above sub-problem is $O(nm)$. Since we need to solve these sub-problems at most $m - 1$ times, for each value of t of the *PMR* decision problem, the total time to compute *PMR* is $O(nm^3)$. Consequently, given m^2 pairs of alternatives, we have:

Theorem 5. *Minimax regret can be computed in $O(nm^5)$ time for the Bucklin voting rule.*

4.5. Egalitarian voting

We have demonstrated above the rather straightforward computation for *PMR* and minimax regret using the independent completion of the partial votes of each voter for decomposable rules (positional scoring), while non-decomposable rules like maximin and Bucklin require the "coordinated completion" of different voters' partial votes. However, certain non-decomposable scoring functions do admit the independent completion of partial votes. Consider the egalitarian voting rule, where $s(a, \mathbf{v}) = \min_i\{m - v_i(a)\}$. While minimizing or maximizing the score of a candidate in partial vote p_i is straightforward, the way in which its *adversarial advantage* is maximized in p_i can depend on other votes. In the cases $Nec_i(w \succ a)$ and $Inc_i(a, w)$, there is only one way to maximize the local advantage of w over a (see above). But when $Nec_i(a \succ w)$, the placement of U either above a or below w influences the egalitarian score of a and w in a way that depends on other votes. However, one can show that unless $PMR(a, w, \mathbf{p})$ is negative, then advantage is maximized by ordering *U below w*. Informally, placing U below w can improve the minimum score of both a and w. However, this placement can only improve the minimum score of a if vote v_i gives a its minimum score over all p_i, in which case the minimum score of w is strictly less than that of a, and $PMR(a, w, \mathbf{p})$ is negative. Since max regret can never be negative, the pair (a, w) cannot define a's max regret. This lets us prove that, unless PMR is negative, $PMR(a, w, \mathbf{p})$ is maximized by ordering U below w in any p_i where $Nec_i(a \succ w)$. The running time for PMR is $O(nm)$ and for MMR is $O(nm^3)$, as in the case of positional scoring rules, since we only need to identify the relevant sets in Fig. 2.

Theorem 6. *Minimax regret can be computed in $O(nm^3)$ time for egalitarian voting.*

5. Preference elicitation for single-winner problems

The ability to compute regret-minimizing winners is critical when only partial information is available. However, in many cases, one has the option of eliciting additional preference information from the voting population in order to reduce minimax regret and improve decision quality. This can be especially important when the minimax regret of the partial profile exceeds some (problem- or domain-specific) tolerance, indicating the potential for divergence from optimality beyond an acceptable threshold.

Of course, as discussed above, elicitation can be difficult with respect to both computation and communication complexity in the worst case. Despite this, we show how minimax regret can be used effectively to guide the elicitation process, providing strong results in practice. As a measure of solution quality, minimax regret can be used to terminate elicitation whenever regret falls to some suitable level (including zero if optimality is desired).[16] More importantly, the solution to the minimax optimization can guide the selection of queries, and voters to whom those queries should be directed, so that an (approximate or exact) optimal solution can often be found quickly. In this section, we describe a simple heuristic strategy to do just this. In Section 5.1 we focus on linear positional (Borda-like) rules while Section 5.2 develops an elicitation strategy for the egalitarian rule. Both algorithms use two specific query types; but these ideas generalize to other rules and other forms of queries.

5.1. Current solution strategy for positional scoring rules

We consider a general technique, called the *current solution strategy (CSS)*, that has been used effectively in both single-agent recommendation systems [16,23,92], multi-agent mechanism design [50–52], and in multi-agent stable matching [41] settings (the latter uses the completion techniques we develop in this paper). Intuitively, CSS identifies voter preference information that helps assess the relative degree of preference between the minimax optimal solution and the adversarial witness within each voter's partial preference ranking, and queries a voter whose response has the greatest potential to reduce the advantage of the witness over the minimax optimal alternative.

CSS is a meta-strategy for elicitation. Its precise instantiation depends on the types of preference queries permitted and the specific voting rule in question. We consider two forms of queries in this work. A *comparison query* identifies a voter k, and asks k to compare two alternatives: "Is $a \succ_k b$?". A *top-t query* identifies and asks voter k to state which alternative is tth in their ranking (we assume that the first $t - 1$ alternatives have already been articulated by k). We describe CSS in detail using comparison queries, but we show how the same intuitions can be adapted to the selection of top-t queries.

Suppose we are allowed to ask any voter a pairwise comparison query. CSS generates queries by considering the current solution to the minimax optimization—i.e., the minimax optimal alternative a and adversarial witness w—and using this to choose a voter-query pair with greatest potential to reduce minimax regret. Notice that if the advantage of w over a is not reduced in some partial vote p_k in response to a query, $PMR(a, w)$ will not change, thus, unless the response changes the minimax optimal solution, MMR will not change. So CSS selects queries that tackle this gap directly.

In this section we focus on positional scoring rules, illustrating the main concepts first using the Borda PSF. We determine the value of posing a query to voter k by considering the three cases in Fig. 2, in each case determining the query with the largest potential reduction given a positive response by k:

Case 1, where $Nec_k(a \succ w)$. Recall that a worst-case completion must place (all alternatives in) set D below w, set C above a, and set U either below w or above a. We can reduce $PMR(a, w)$ by asking two different types of queries: $d \succ w$

[16] If determining *termination* for an elicitation procedure is hard for a particular voting rule (i.e., determining whether a winner is known given elicited preferences) [33,94], then so is computing minimax regret; or equivalently, if computing minimax regret is easy (as demonstrated for certain rules above), so is termination.

for some $d \in D$ or $a \succ c$ for $c \in C$. In each case, a positive response will position alternatives between a and w, thus reducing $PMR(a, w)$ by increasing the (worst-case) position of a relative to w in p_k, Specifically, a positive response to $d \succ w$ prevents the adversary from placing d, or any of its ancestors in D, below w in the completion. And a positive response to $a \succ c$ prevents the adversary from placing c, or any of its descendants in D, above a in the completion. We pick the alternative in $C \cup D$ (and corresponding query) with greatest potential to reduce *PMR* in the case of such a positive response. For linear scoring rules, this potential is measured by the number of ancestors of d in set D (all of which will be positioned between a and w if d is), and the number of descendants of c in set C. If $C \cup D = \emptyset$, we can ask two other query types, $u \succ w$ or $a \succ u$ for some $u \in U$. These do not reduce *PMR* directly, but shift u and its ancestors in U to set C (for query $u \succ w$) or u and its descendants in U to set D (for query $a \succ u$). The u with the potential to shift greatest number of alternatives within U to some other set is chosen.

Case 2, where $Nec_k(w \succ a)$. Recall that a worst-case completion must place sets E, A and U between w and a. We can reduce $PMR(a, w)$ by asking four different types of queries: $a \succ e$ for some $e \in E$; $a \succ u$ for $u \in U$; $x \succ w$ for some $x \in A$; or $u \succ w$ for some $u \in U$. A positive response to any such query will reduce *PMR*. In the case of query $a \succ e$ (respectively, $a \succ u$), it increases the (worst-case) score of a in p_k, by preventing the adversary from positioning e (respectively, u), or any of its descendants in E (respectively, U), between w and a. Similarly, in the case of query $x \succ w$ (respectively, $u \succ w$), it reduces the (worst-case) score of w in p_k, by preventing the adversary from positioning x (respectively, u), or any of its ancestors in A (respectively., U) between w and a. Selection is again made by choosing the alternative and query that have the greatest potential (i.e., the number of alternatives that are removed from their respective sets E, A or U in case of a positive response).

Case 3, where $Inc_k(a, w)$. We can reduce $PMR(a, w)$ by asking several different queries, however, heuristically, we always choose to ask if $a \succ w$, since a positive response reverses p_k's contribution to *PMR* from positive to negative. *Any* response will move partial vote p_k into either case 1 or case 2.

We note that each voter is "scored" by assessing which query has the greatest potential to reduce adversarial advantage in her partial vote, based on the classification of that partial vote into one of the three cases above. When asking a single query at each round, we choose the voter whose corresponding best query has the greatest potential (breaking ties arbitrarily).

CSS must eventually terminate with an optimal solution:

Proposition 7. *Unless $MMR(\mathbf{p}) = 0$, CSS will always select a voter k and comparison query $a_i \succ_k a_j$ such that $Inc_k(a_i, a_j)$. That is, CSS never generates redundant queries and eventually outputs a winner under the positional scoring rule.*

Proof. Suppose that $MMR(\mathbf{p}) > 0$, that a is a minimax optimal alternative, and w is its adversarial witness. If there exists a partial vote p_k such that $Inc_k(a, w)$, then CSS generates the query $a \succ_k w$ for voter k. If there is a partial vote p_k such that $Nec_k(a \succ w)$ and at least one of the sets D, C or U (see Fig. 2) is non-empty, then CSS will generate a (non-redundant) query for voter k (see Case 1 of CSS). If there is a partial vote p_k such that $Nec_k(a \prec w)$ and at least one of the sets E, A or U (see Fig. 2) is non-empty, then again CSS will generate a (non-redundant) query for voter k (see Case 2 of CSS).

Finally, suppose no partial vote satisfies any of the three conditions above. We argue this is impossible unless minimax regret is zero. If no partial vote satisfies any of the three conditions, this implies that all partial votes fall into Case 1, where $Nec(a \succ w)$, or Case 2, where $Nec(a \prec w)$. Furthermore, all Case 1 votes satisfy $D \cup C \cup U = \emptyset$ and all Case 2 votes satisfy $E \cup A \cup U = \emptyset$. However, this means the exact rank positions and hence the positional scores of a and w can be determined in every partial vote (see Fig. 2). By definition of *MMR*, w must have a strictly higher score than a in *some* full vote completion if $MMR(\mathbf{p}) > 0$. Therefore, since the positional scores of a and w are fully known in all partial votes, it follows that w would have a strictly higher score than a in *every* completion of $\mathbf{p}$, contradicting the minimax optimality of a (in particular w would have a strictly lower max regret than a). □

CSS can be adapted, using similar intuitions, to generate top-t queries. Such queries are asked of each voter in order—no voter is asked for the second-ranked candidate before their revealing their first-ranked candidate, their third before their second, etc. This means that CSS need only select a voter at any stage—the query is determined once the voter is selected. If elicitation is confined to top-t queries, for a given voter we consider four cases to assess the potential reduction in *MMR* associated with querying that voter (again assuming minimax optimal a and adversarial witness w):

Case 1. If a and w are already in the voter's top-t list, then their scores are known with certainty and $PMR(a, w)$ cannot be reduced by querying this voter.

Case 2. If a is in the voter's top-t list but w is not, asking for the next highest alternative—assuming optimistically that the response is not w—reduces the adversarial advantage by one (i.e., the adversary would place w at rank $t + 2$ instead of $t + 1$).

Case 3. If a is not in top-t list, but w is, the adversary maximizes *PMR* by placing a at the bottom of the ranking. If a is (optimistically) returned as the next highest alternative, *PMR* is reduced by $m - (t+1)$ (since a must be at position $t+1$).

Case 4. If neither a nor w are in the top-t list, the adversary will place a at the bottom of the ranking and w at position $t+1$. If a is (optimistically) returned as the next highest alternative, *PMR* is reduced by $m - t$ (a must now be placed at position $t+1$ and w at $t+2$).

In each of the cases above, we calculate the potential *PMR* reduction for each voter, and query the voter with the largest potential reduction. It is clear that, unless minimax regret is zero, CSS will always have some voter to query with positive potential reduction.

5.2. Current solution strategy for egalitarian voting

As discussed above, CSS is a meta-strategy that requires specific instantiations for different voting rules. While we focused on CSS for positional scoring above, we illustrate the generality of CSS by developing a CSS strategy for the egalitarian rule.

Using CSS for egalitarian voting relies on similar intuition as for positional scoring rules, but must deal with potential interaction across different profiles (as in the case of computing *MMR* for egalitarian in Section 4.5. Suppose we have minimax regret winner a and its witness w. Again, the key idea is to reduce the pairwise advantage of w with respect to a in adversarial completions of the partial votes. Thus we focus on asking queries involving a or w that have the greatest potential to reduce their pairwise max regret. We consider each voter k and the corresponding potential reduction in $PMR(a, w)$, again focusing on three main cases:

Case 1, where $\boldsymbol{Inc_k(a, w)}$. We measure the impact of query $a \succ w$. The greatest potential regret reduction comes with an affirmative response. This occurs in one of two ways. (a) If the resulting configuration has w decreasing its rank in vote k such that its overall egalitarian score decreases (i.e., partial vote p_k implies that w be ranked lowest among all votes in an adversarial completion, which we can obtain via the *MMR* algorithm for egalitarian). (b) If a increases its overall egalitarian score (i.e., prior to query response, voter k ranks a lowest amongst all voters in an adversarial completion, but a is elevated in k's ranking after the response). In either case, the query has a strictly positive impact in reducing pairwise regret of a and w.

Case 2, where $\boldsymbol{Nec_k(w \succ a)}$. Consider all queries in the corresponding Case 2 of the CSS algorithm for the positional scoring rules. Each such query (i.e., for $x \in A$, $w \succ x$; or for $x \in E$, $x \succ a$), may either reduce w's egalitarian score or increase a's score, depending on whether partial vote p_k is the unique vote in an adversarial preference profile completion that ranks w or a the lowest. Of all such queries, we select that with the highest potential *PMR* reduction. This can be done by first identifying whether p_k is the unique partial vote whose worst-case completion (as computed by the egalitarian *MMR* algorithm) ranks w highest or a lowest among all partial vote completions—if so, the above queries can potentially reduce regret.

Case 3, where $\boldsymbol{Nec_k(a \succ w)}$. Consider the similar situation for queries as in Case 2. These queries can potentially reduce the overall score of w or increase the score of a, depending on whether p_k is the unique optimizer the egalitarian score of either a or w.

We run a potential regret analysis for each voter and query pair, and select the pair with largest such potential reduction in pairwise max regret. Note that the potential to reduce *PMR* might be zero for any single pairwise query—in such cases we can either ask a random query or generate a query that can help reduce regret in later rounds. When this case occurs, it implies that multiple partial votes minimize (respectively, maximize) the position of a (respectively, w). That is, one of the two following conditions must hold:

1. There are two or more votes that under the worst-case completion minimize the position/score of a, or
2. There are two or more votes that under the worst-case completion maximize the position/score of w.

Even though a single query cannot reduce *PMR*, any query that improves the rank of a (pairwise comparison queries that involve a) or worsens the rank of w (comparisons that involve w) in any such vote is helpful in reducing *PMR* in later rounds.

5.3. Empirical evaluation

Our experiments focus on evaluating elicitation strategies with respect to the Borda rule, but our conclusions give us general intuitions about the advantages of CSS over some baseline strategies. As benchmarks in our experiments, we use two other strategies to contrast with CSS. The *random strategy (Rand)* randomly chooses a voter k and a comparison query such that $Inc_k(a_i, a_j)$ (so the query response always bears information). With top-t queries, Rand only needs to choose voter

k at random. The *volumetric strategy (Vol)* selects a voter k and query $a_i \succ a_j$ that maximizes the number of *new* pairwise preferences revealed (given the worst response):

$$Vol(p_k) = \max_{a_i, a_j} \min \left\{ \begin{array}{l} |tc(p_k \cup \{a_i \succ a_j\})|, \\ |tc(p_k \cup \{a_j \succ a_i\})| \end{array} \right\},$$

where tc denotes transitive closure. This strategy reduces preference uncertainty maximally, without regard for the "relevance" of the revealed preference information to winner determination (much like volumetric strategies for in polyhedral conjoint analysis for single-agent problems [91].) Its application to top-t queries involves selecting the voter whose next-ranked candidate reduces uncertainty the most. Since this voter *must* be one who has ranked the fewest candidates, the strategy reduces to a simple sequential iteration: each voter in turn is asked for their top-ranked candidate; then each is asked for their second-ranked candidate; and so on. We refer to Vol in this case as SequentialTop.

In previous work (see Section 2.4), Kalech et al. [53] proposed two vote elicitation algorithms. Their first method is essentially the SequentialTop method described above, and proceeds in rounds in which each voter is queried for their next most preferred choice. It uses necessary winner computation for termination. This contrasts with our CSS approach, which is much more subtle and incremental: we identify a *particular voter* to query at each stage by evaluating its potential to reduce minimax regret. We see in our experiments that this can reduce the number of required queries substantially. Furthermore, our elicitation methods are anytime: querying can terminate when minimax regret is sufficiently small, and we show below that this further reduces the number of queries significantly.

Their second algorithm proceeds for a predetermined number of rounds, asking each voter at each stage for fixed number of positional rankings. Since termination is predetermined, necessary winners may not result (instead possible winners are returned), and interesting tradeoffs between the number of rounds and amount of information per round are explored. As the authors acknowledge, this scheme provides no guarantee of winner optimality or any bounds on quality. A key advantage of our minimax regret-based scheme is that a natural, precise objective is being minimized, and anytime quality guarantees are provided.[17] Their approach does, however, batch queries, so voters are only queried a few times, though each query may request more information than the CSS scheme. This batching can minimize user interruption as well as user *latency* (since voters are not required to wait until the responses of other voters are delivered before their next query is received). We discuss the advantages of batching—and how minimax regret can be adapted to handle batching—in Section 8.

We test CSS on three different data sets:

- *Sushi*: The Sushi data set [54] contains 5000 full preference rankings over 10 varieties of sushi;
- *Irish*: The Irish data set comprises the votes of the 2002 Irish national election from the Dublin North constituency. It contains 43,942 ballots of the top-t form over 12 candidates. We use the subset consisting of the 3662 complete ballots (i.e., that rank all 12 candidates).[18]
- *Mallows*: We generate random vote profiles, each with 100 random rankings over 20 alternatives, where each ranking is drawn from a *Mallows ϕ-model* [68], using several different parameter settings (as we explain further below). The Mallows model is a distribution over rankings given by a modal ranking σ and dispersion parameter $\phi \in (0, 1]$ with $\Pr(r|\sigma, \phi) \propto \phi^{d(r,\sigma)}$, where d is Kendall's τ-distance. Smaller ϕ concentrates probability mass around the modal ranking σ (i.e., all voters have nearly identical rankings), while $\phi = 1$ gives the uniform distribution over rankings, also known as *impartial culture*.

These data sets are used to generate responses to elicitation queries, and reflect different forms of group decisions, including political voting and applications related to recommender systems.

We test CSS on each data set, measuring how quickly minimax regret reduces as a function of the total number of queries asked of the voting population. For the real-world data sets (Sushi, Irish), we use both pairwise comparisons and top-t queries, and compare the performance of CSS to both the random and volumetric elicitation strategies. On the Mallows (random) data, we use only pairwise comparisons and analyze the impact on convergence as we vary the dispersion parameter. In all cases, we use Borda scoring to measure candidate quality (similar results hold for other positional scoring rules).

The top two plots in Fig. 4 show *MMR* as a function of the total number of queries asked on the Sushi and Irish data sets. We see that CSS offers superior elicitation performance with both pairwise comparisons and top-t queries. With Sushi, CSS reaches the optimal solution (i.e., the provable winner with $MMR = 0$) after an average of only 11.82 comparison queries per voter. This compares with 20.64 queries for Vol and 20.63 queries for Rand (as well as the 25 queries required by the theoretically optimal MergeSort to determine full voter rankings). With top-t queries, CSS needs only 3.40 queries per voter to reach $MMR = 0$, compared to 4.18 for Seq, and 5.50 for Rand. With Irish, results are similar: CSS reaches optimality with 18.57 comparison queries and 5.47 top-t queries per voter. This stands in contrast with 31.82 comparisons and 6.91 top-t queries for Vol/Seq; 31.22 comparisons and 8.38 top-t queries for Rand, and 33 comparisons for MergeSort. Note that top-t

[17] We also note that elicitation of pairwise preferences is not considered in [53]; such queries are extremely valuable and arise naturally in many domains such as search, information retrieval, consumer product comparisons, etc.

[18] See www.dublincountyreturningofficer.com.

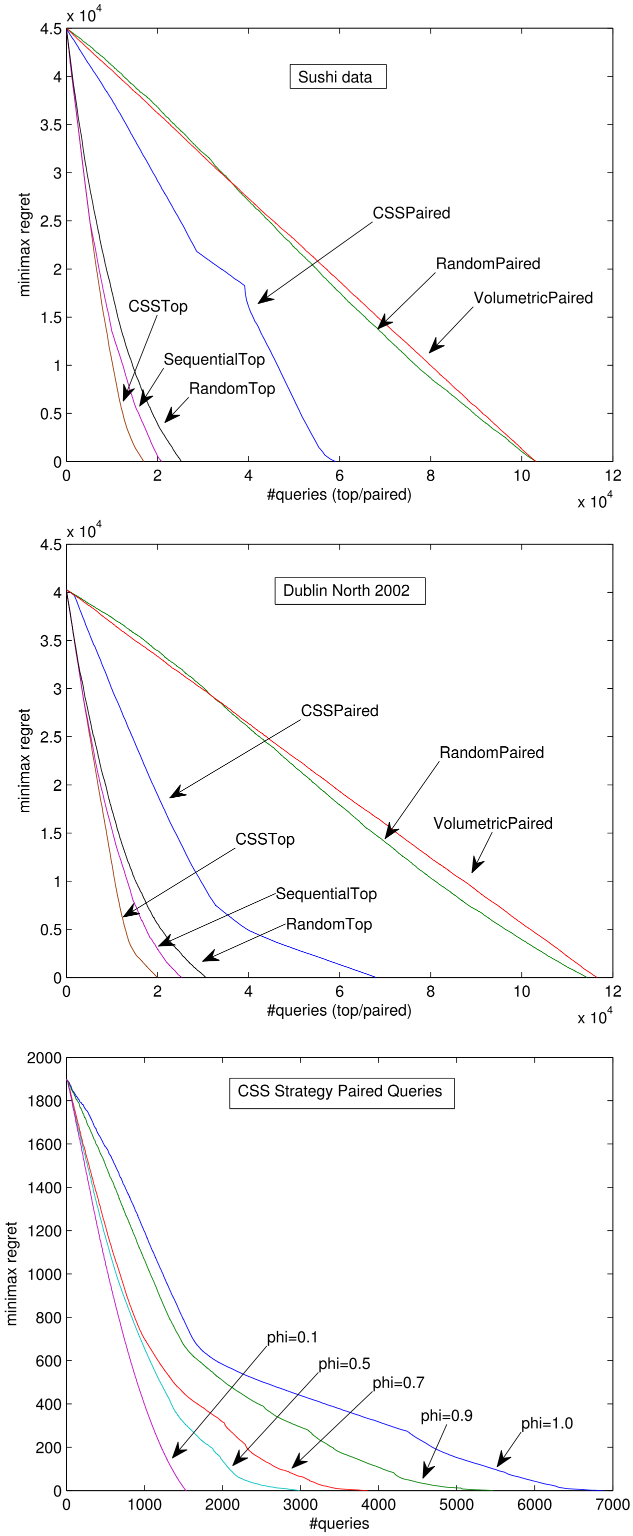

Fig. 4. Performance of elicitation algorithms (paired and top queries) on Sushi, Dublin and Mallows data. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

queries are "information rich" as they provide many pairwise comparisons per response.[19] Thus, while CSS's advantage is somewhat smaller in the top-t case (though RandomTop still requires over 50% more queries to reach $MMR = 0$ in Irish), the fact that there is an advantage is of greater significance due to the greater "intensity" of each query.

Critically, if one is interested in approximate solutions, we see that CSS reduces *MMR* very quickly, providing high-quality solutions after very few queries. For example, with Irish, CSS reduces *MMR* to 18% of its initial value (with no voter preference data) after only 5.82 comparison queries per voter, which is a small fraction of the queries required to elicit full rankings. By contrast, to reduce regret to the same degree requires 25.77 comparisons for Vol, and 24.03 comparisons for Rand. Computationally, for problems of this size, CSS takes only a few milliseconds on average (wall clock time) to find the best agent/comparison query (including time needed to recompute the *MMR*-solution).

On the synthetic Mallows data set, we sample 100 complete voter profiles, each with 100 voters and 20 alternatives, for each of several different values of ϕ. The same impact of CSS on *MMR* as a function of the number of pairwise comparisons asked is shown in the bottom plot of Fig. 4. With larger ϕ, more queries are clearly needed to reach the same level of regret, which conforms to our intuitions that intelligent elicitation schemes can take significant advantage of less uniform preferences to minimize queries and voter effort (and conversely, that with almost uniformly random preferences, nearly full rankings must be obtained). Work in behavioral social choice strongly suggests that real-world preferences are not uniformly random [81], and CSS seems to perform especially well in this case; indeed our results on Sushi and Irish suggest that real preferences are not uniform, and contain regularities that can be readily exploited to reduce the informational complexity of voting.

6. Robust slate optimization for multi-winner problems

We now turn our attention to the multi-winner problem, focusing primarily on the Chamberlin-Courant rule, and consider the robust optimization of a *slate* of alternatives given a partial preference profile, using minimax regret as our robustness criterion. We begin in Section 6.1 by defining minimax regret for multi-winner problems. We then discuss the computation of minimax optimal slates in Section 6.2, focusing our attention on linear positional scoring rules (like Borda) for ease of exposition. We describe the relevant completion principles, and show that the computational problem is hard in general—it is in complexity class XP with respect to K, and can be practically solved for small, bounded slate sizes K. In Section 6.3, we describe a tractable greedy algorithm for approximating minimax-optimal slates and derive relevant approximation ratios. We explore the use of these methods to drive preference elicitation for multi-winner problems in the following section.

6.1. Minimax regret for slate optimization

We consider the slate optimization problem described in Section 2.3 in which our goal is to select, given a vote profile, a set or *slate of alternatives* $\bar{a} \subseteq A$ of up to K alternatives that maximizes total voter satisfaction. We assume that voter satisfaction with candidates is given by some scoring function induced by their ranking (e.g., the Borda PSF), and that a voter's satisfaction with the slate is given by *the score of their most preferred alternative on the slate*. Recalling Eqs. (1) and (2), we define the *score* of a K-set and the *optimal K-set* as follows:

$$S(\bar{a}, \mathbf{v}) = \sum_{i \in N} S_i(\bar{a}) \quad \text{where} \quad S_i(\bar{a}) = \max_{a \in \bar{a}} \alpha_i(a), \tag{14}$$

$$\bar{a}^*_{\mathbf{v}} = \operatorname*{argmax}_{|\bar{a}| \le K} S(\bar{a}, \mathbf{v}). \tag{15}$$

We suppress dependence of S on α since the PSF will be fixed and clear from context. This corresponds to the Chamberlin-Courant rule [26] when α is the Borda PSF. For more general α, this is known as the *limited choice model* [62].

We now consider the problem of selecting an optimal K-set of alternatives when we have only a partial preference profile $\mathbf{p}$ rather than a complete profile $\mathbf{v}$. We again adopt *minimax regret* as a robustness criterion for making decisions with a partial profile. The definitions of pairwise maximum regret, maximum regret, and minimax regret are analogous to those in the single-winner case, with decisions (and adversarial choices) corresponding to slates of K options rather than single alternatives. We first present definitions (for sets of fixed size K) then explain the intuitions. Recall that $\bar{a}$ and $\overline{w}$ represent slates of options.

$$Regret(\bar{a}, \mathbf{v}) = \max_{|\overline{w}| \le K} S(\overline{w}, \mathbf{v}) - S(\bar{a}, \mathbf{v}) \tag{16}$$

$$PMR(\bar{a}, \overline{w}, \mathbf{p}) = \max_{\mathbf{v} \in C(\mathbf{p})} S(\overline{w}, \mathbf{v}) - S(\bar{a}, \mathbf{v}) \tag{17}$$

[19] We expect that, in general, answering a top-t query will impose a lower cognitive burden on the user (and be less time-consuming) than the equivalent collection of pairwise comparison queries due to the overhead of assessing to each pairwise comparison in isolation. Of course, the top-t strategy requires users to answer informationally rich queries even when not all of the implied comparisons are necessary for winner determination.

Table 1
Example of computing max regret for each candidate slate $\bar{a}$ of size 2. The notation $v_i : +\{\dots\}$ indicates additional pairwise comparisons added to p_i for the adversarial completion. The minimax optimal slate is $\bar{a}^*_{\mathbf{p}} = \{a_1, a_4\}$ with max regret of 2.

Slate $\bar{a}$	Adversarial completion	Witness $\overline{w}$	Max regret $MR(\bar{a}, \mathbf{p})$
$\{a_1, a_2\}$	$v_1, v_2 : +\{a_4 \succ a_1\}$ $v_3 : +\{a_1 \succ a_2\}$	$\{a_3, a_4\}$	$9-6=3$
$\{a_1, a_3\}$	$v_1, v_2 : +\{a_4 \succ a_1\}$ $v_3 : +\{a_4 \succ a_3\}$	$\{a_2, a_4\}$	$9-5=4$
$\{a_1, a_4\}$	$v_1, v_2 : +\{a_3 \succ a_4\}$ $v_3 : +\{a_2 \succ a_3, a_1 \succ a_4\}$	$\{a_1, a_2\}$	$9-7=2$
$\{a_2, a_3\}$	$v_1, v_2 : +\{a_4 \succ a_1\}$ $v_3 : +\{a_4 \succ a_3\}$	$\{a_2, a_4\}$	$9-5=4$
$\{a_2, a_4\}$	$v_1, v_2 : +\{a_3 \succ a_4\}$ $v_3 : +\{a_1 \succ a_2\}$	$\{a_1, a_3\}$	$9-5=4$
$\{a_3, a_4\}$	$v_1, v_2 : +\{a_3 \succ a_4\}$	$\{a_1, a_2\}$	$9-4=5$

$$MR(\bar{a}, \mathbf{p}) = \max_{\mathbf{v} \in C(\mathbf{p})} Regret(\bar{a}, \mathbf{v})$$

$$= \max_{|\overline{w}| \le K} PMR(a, \overline{w}, \mathbf{p}) \tag{18}$$

$$MMR(\mathbf{p}) = \min_{|\bar{a}| \le K} MR(\bar{a}, \mathbf{p}) \tag{19}$$

$$\bar{a}^*_{\mathbf{p}} \in \operatorname*{argmin}_{|\bar{a}| \le K} MR(\bar{a}, \mathbf{p}) \tag{20}$$

Given a vote profile $\mathbf{v}$, $Regret(\bar{a}, \mathbf{v})$ describes the loss in satisfaction associated with offering set $\bar{a}$ rather than the optimal K-set. Given a partial profile $\mathbf{p}$, the *pairwise max regret* $PMR(\bar{a}, \overline{w}, \mathbf{p})$ is the worst-case loss that could be incurred, under all possible realizations of voter preferences, by offering $\bar{a}$ rather than $\overline{w}$. Note that our definition of *PMR* does not impose constraints on set sizes, a fact we exploit below. The *max regret* $MR(\bar{a}, \mathbf{p})$ of set $\bar{a}$ is the worst-case loss relative to the *optimal* K-set under all preference realizations: this bounds the loss associated with $\bar{a}$ given our preference uncertainty. Finally, a *minimax optimal set* $\bar{a}^*_{\mathbf{p}}$ is one with minimum max regret or *minimax regret* $MMR(\mathbf{p})$.

Observation 8. If $MMR(\mathbf{p}) = 0$, then $\bar{a}^*_{\mathbf{p}}$ is an optimal slate of alternatives for any $\mathbf{v} \in C(\mathbf{p})$.

To illustrate the various regret-based optimization concepts above, consider the following example.

Example 9. Suppose we have four alternatives $A = \{a_1, a_2, a_3, a_4\}$, and three voters with partial preferences $p_1 = p_2 = \{a_1 \succ a_2 \succ a_3\}$ and $p_3 = \{a_3 \succ a_1, a_2 \succ a_4\}$. We use Borda scoring, with $\alpha(i) = 4 - i$ for $i = 1, \dots, 4$. Assume we are interested in slates of size $K = 2$ (we can also compute max regret for singleton slates but the calculations are similar so we omit for brevity). Table 1 shows the max regret calculations for each slate of size 2.

For example, for candidate slate $\bar{a} = \{a_1, a_2\}$, to maximize regret the adversary ranks a_4 above a_1 for the first two voters (achieving regret of 2 when a_4 is in the witness slate) and a_1 above a_2 for the third voter (achieving regret of 1 when a_3 is in the witness slate). This results in max regret of 3 for $\bar{a}$. Max regret for the other candidate slates is derived similarly. The minimax optimal slate is $\{a_1, a_4\}$. To reduce minimax regret, one can ask specific preference queries of particular voters, as we elaborate in the next section. For example, if voter 3 is asked whether $a_1 \succ a_2$, a positive response reduces the max regret of slate $\{a_1, a_4\}$ by 1—it remains minimax optimal, but becomes closer to being provably optimal. (The query $a_4 \succ a_3$ has a similar effect.)

6.2. Computing minimax optimal slates

Before discussing computation of minimax regret, we begin with the simpler problem of computing pairwise max regret. From Eqs. (17)–(19), we see that the regret-optimal slate $\bar{a}^*_{\mathbf{p}}$ can be determined by first computing $PMR(\bar{a}, \overline{w}, \mathbf{p})$ for all pairs of K-sets $\bar{a}, \overline{w}$, followed by maximizing over $\overline{w}$ to determine $MR(\bar{a}, \mathbf{p})$, and then minimizing over these terms to compute $MMR(\mathbf{p})$. If K is small, then robust optimization is efficient if *PMR* can be computed effectively, a problem to which we now turn. (We discuss an approach for large K in the next subsection.) Just as with single-winner problems using scoring rules, one can show *PMR* is additively decomposable:

$$PMR(\bar{a}, \overline{w}, p_i) = \max_{v_i \in C(p_i)} S(\overline{w}, v_i) - S(\bar{a}, v_i),$$

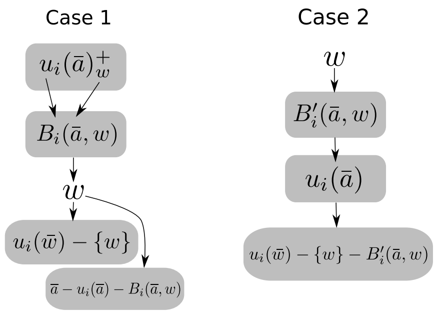

Fig. 5. Adversarial completions of p_i regret $PMR(u_i(\bar{a}), \{w\}, p_i)$, Cases 1 and 2.

$$PMR(\bar{a}, \overline{w}, \mathbf{p}) = \sum_{i \in N} PMR(\bar{a}, \overline{w}, p_i)\ . \tag{21}$$

Thus we can compute the contributions of each voter i to *PMR* independently. When i is presented with slate $\bar{a}$, she will choose her most preferred alternative from $\bar{a}$, and similarly for slate $\overline{w}$. Define the *undominated elements* for voter i in any set $\bar{a}$ to be:

$$u_i(\bar{a}) = \{a \in \bar{a} : \nexists a' \in \bar{a} \text{ s.t. } a' \succ_i a\}.$$

If presented with a slate $\bar{a}$, i's maximal satisfaction—informally, we call this i's *choice* for $\bar{a}$—can only be derived from one of these undominated elements, no matter what completion of i's partial preferences that reflect her true underlying preferences. In the ranking $v_i \in C(p_i)$ that maximizes pairwise regret, only one element in $\overline{w}$ will be chosen by i (the most preferred), which defines *PMR*:

$$PMR(\bar{a}, \overline{w}, p_i) = \max_{w \in u_i(\overline{w})} PMR(u_i(\bar{a}), \{w\}, p_i). \tag{22}$$

Given this, there are two cases to consider when determining the adversarial completion $v_i \in C(p_i)$ that maximizes $PMR(u_i(\bar{a}), \{w\}, p_i)$.

Case 1. Suppose there is an $a \in \bar{a}$ such that $Nec_i(a \succ w)$. This means there is no completion in which i would choose w, so *PMR* is negative. Maximizing pairwise regret requires reducing the "gap" between the most preferred $a^* \in \bar{a}$ and w. The only alternatives that must lie between the most preferred a and w are undominated elements $u_i(\bar{a})$ of $\bar{a}$ that dominate w, or those b known to lie between such an a and w. Define

$$u_i(\bar{a})^+_w = \{a \in u_i(\bar{a}) : a \succ_i w\},$$
$$B_i(\bar{a}, w) = \{b \in A : \exists a \in u_i(\bar{a})^+_w, a \succ_i b \succ_i w\}.$$

$B_i(\bar{a}, w)$ includes all alternatives that must lie between the best $a \in \bar{a}$ and w (the specific choice or placement of the elements in these two sets has no impact on *PMR*). Every other alternative can consistently be ordered above the best a or below w depending on constraints in p_i.[20] Thus we have:

$$PMR(u_i(\bar{a}), \{w\}, p_i) = -|u_i(\bar{a})^+_w| - |B_i(\bar{a}, w)|.$$

See Fig. 5 (Case 1) for an illustration for an illustration of the different sets defined above and their relationship, as well as a schematic view of the completion.

Case 2. Now suppose that for voter i, no element in $a \in \bar{a}$ is known to be preferred to w. If $w \in u_i(\bar{a})$ then $PMR(u_i(\bar{a}), \{w\}, p_i) = 0$, since any adversarial completion can place w above all alternatives in $u_i(\bar{a}) \setminus \{w\}$ (otherwise regret would be negative). Otherwise the desired completion must maximize the gap between w and any alternative in $u_i(\bar{a})$. The following options can be placed between w and $\bar{a}$:

$$B'_i(\bar{a}, w) = \{b \in A \setminus \bar{a} : b \not\succ_i w \text{ and } \forall a \in u_i(\bar{a}), a \not\succ_i b\}. \tag{23}$$

The relative ordering of these alternatives does not impact regret. With $B'_i(\bar{a}, w)$ placed below w, some alternative from $u_i(\bar{a})$ must lie immediately below the last element of this set (becoming the most preferred $a \in \bar{a}$). Thus, we have:

[20] For "nonlinear" scoring rules, where the score difference for two options depends not just on relative, but also absolute rank position, placement of options above or below a and w requires more care, but is straightforward in most cases.

$$PMR(u_i(\bar{a}), \{w\}, p_i) = \begin{cases} 1 + |B'_i(\bar{a}, w)| & \text{if } w \notin u_i(\bar{a}), \\ 0 & \text{otherwise.} \end{cases}$$

See Fig. 5 (Case 2) for an illustration (where $w \notin u_i(\bar{a})$).

In both cases, the undominated sets $u_i(\bar{a})$ and $u_i(\overline{w})$ can be computed in $O(K^2)$ time. In case 1, $u_i(\bar{a})^+_w$ can be computed in $O(K)$ time once $u_i(\bar{a})$ is known, and $B_i(\bar{a}, w)$ can be computed in $O(mK)$ time by checking if each $b \in A$ satisfies the constraints with respect to $u_i(\bar{a})^+_w$ and w. For case 2, $B'_i(\bar{a}, w)$ can be found in time $O(mK)$ by checking each $b \in A$ with w and the alternatives in $u_i(\bar{a})$. Using Eqs. (22) and (21), $PMR(\bar{a}, \overline{w}, \mathbf{p})$ can be computed in $O(n(K^2 + K(K + mK))) = O(nmK^2)$ time. Note that for $K = 1$, the approach is identical to *PMR* computation for the single-winner Borda rule. Putting this together we have:

Theorem 10. *$PMR(\bar{a}, \overline{w}, \mathbf{p})$ is given by:*

$$\sum_{i \in N} \max_{w \in u_i(\overline{w})} \begin{cases} -|u_i(\bar{a})^+_w| - |B_i(\bar{a}, w)| & \text{if } \exists a \in \bar{a} : a \succ_i w, \\ 1 + |B'_i(\bar{a}, w)| & \text{otherwise, and } w \notin u_i(\bar{a}), \\ 0 & \text{otherwise,} \end{cases}$$

and is computable in $O(nmK^2)$ time.

The minimax optimal slate $\bar{a}^*_{\mathbf{p}}$ can be constructed by computing max regret $MR(\bar{a}, \mathbf{p})$ for each slate $\bar{a}$ of size K, and then selecting the slate $\bar{a}^*_{\mathbf{p}}$ that minimizes max regret. In turn, $MR(\bar{a}, \mathbf{p})$ can be computed by determining the *PMR* of slate $\bar{a}$ for each witness set $\overline{w}$ of size K. Hence:

Proposition 11. *The minimax regret optimal slate $\bar{a}^*_{\mathbf{p}}$ can be computed in time $O(nm^{2K+1}K^2)$.*

The additive decomposability of *PMR* has the nice computational consequence that, during the course of incremental elicitation (see Section 7), one need only update the contributions to *PMR* of those agents who have their partial preferences updated by responding to a query. For slates of small bounded size K, enumeration of alternative sets may be practical. Indeed, if K is bounded, minimax optimal slates can be computed in polynomial time (in n and m). In other words, the problem is in the parameterized complexity class XP with respect to slate size K.[21]

However, in general, since this form of multi-winner selection and budgeted social choice is an NP-hard optimization problem [62] (as are variants [78]), finding the minimax optimal slate is also NP-hard (simply let $\mathbf{p}$ be a full preference profile). Indeed, simply computing $MR(\bar{a}, \mathbf{p})$ is NP-complete:

Theorem 12. *Given threshold $r \geq 0$, partial profile $\mathbf{p}$, set size K, and set $\bar{a}$ of size at most K, deciding if $MR(\bar{a}, \mathbf{p}) \geq r$ (i.e., does some set $\overline{w}$ of size at most K satisfy $PMR(\bar{a}, \overline{w}, \mathbf{p}) \geq r$) is NP-complete.*

Proof. The decision problem is in NP since a witness certificate is a set $\overline{w}$, for which we can compute the pairwise max regret in polynomial time (Theorem 10) and hence check if $PMR(\bar{a}, \overline{w}, \mathbf{p}) \geq r$.

To show NP-hardness, we reduce from the decision problem defined by the optimization objective given in Eq. (15) using the Borda PSF. This *limited choice (LC)*[22] decision problem is: "given a threshold r', budget K and complete vote profile $\mathbf{v}$, does there exist a slate $\bar{a}'$ of size at most K such that $S(\bar{a}', \mathbf{v}) \geq r'$?" This is known to be NP-hard [62].

Given any LC instance with threshold r', budget K and complete profile $\mathbf{v}$ over a set A of m alternatives, we transform it into a partial profile $\mathbf{p}$ with $m + K$ alternatives (the m original alternatives A plus K "dummy" alternatives D), and we let $\bar{a} = D$ consist of the dummy alternatives. Define each partial vote p_i to be identical to v_i on the top-m ranked alternatives (which are from A), with the remaining dummy alternatives D ranked below, but for which their relative ordering is unspecified. Clearly any completion of the partial votes places the dummy items as the last K ranked items for all the voters, therefore the score of $\bar{a}$ for any completion $\mathbf{v}' \in C(\mathbf{p})$ is $S(\bar{a}, \mathbf{v}') = \sum_{i \in N} \max_{a \in \bar{a}} \alpha_i(a) = n\alpha(m+1) = nK$. In the transformed instance, we set $r = r'$.

A witness $\bar{a}'$ to a positive instance of the LC decision problem implies that $S(\bar{a}', \mathbf{v}) \geq r'$, which in turn ensures we have a positive instance of our max-regret decision problem where $r = r'$. To see this, let witness certificate $\overline{w} = \bar{a}'$. The alternatives in this witness set are ranked in the top m positions in each of the n partial votes p_i (i.e., their rank positions are already fixed and are K positions higher than their rank positions in $\mathbf{v}$, since $\mathbf{p}$ has K additional dummy items). Hence, $PMR(\bar{a}, \bar{a}', \mathbf{p}) = \max_{\mathbf{v}' \in C(\mathbf{p})} S(\bar{a}', \mathbf{v}') - S(\bar{a}, \mathbf{v}') \geq (r' + nK) - nK = r$.

A negative instance of the LC decision problem implies there is no subset $\bar{a}' \subseteq A$ of size at most K such that its score exceeds r'. This implies that, for the max-regret decision problem, there is no witness set $\overline{w}$ of size at most K, where

[21] Betzler at al. [13] have shown that the computation of Chamberlin-Courant winners with full preferences is W[2]-hard with respect to K. This means membership in FPT (i.e., fixed-parameter tractability) is unlikely, thus their XP algorithms are the best one can realistically hope for.

[22] This is also known as Chamberlin-Courant winner determination [26].

$\overline{w} \subseteq A$, such that $PMR(\bar{a}, \overline{w}, \mathbf{p}) \geq r' = r$. If $\overline{w}$ contains any of the dummy alternatives then clearly its score must decrease if that dummy alternative is swapped for an alternative $b \in A$ (since b has a higher rank position), thus the pairwise max regret would only decrease. Hence, the transformed instance is a negative instance for the max-regret decision problem. This completes the reduction, which has polynomial time complexity. □

6.3. A greedy algorithm for robust slate selection

Given the intractability of computing a minimax optimal slate $\bar{a}^*_{\mathbf{p}}$ for a partial profile $\mathbf{p}$ in general, we investigate the possibility of an efficient algorithm for approximating the problem that will be practical even for large values of K. It turns out that a relatively simple greedy optimization procedure can be used for this purpose.

Before describing our method, we note that the slate optimization problem in the full information case (i.e., when all voter preferences are fully known) is submodular [62]. Intuitively, this holds because, for any two slates X, Y where $X \subseteq Y$, adding an alternative x to X results in a subset N_x of voters who choose x as their preferred element of X. Adding x to Y results in the corresponding subset N'_x of voters who choose x from Y. Since $X \subseteq Y$, it must be that $N'_x \subseteq N_x$ (anyone who chooses x from $Y \cup \{x\}$ must also do so from $X \cup \{x\}$; furthermore, the marginal score of x relative to slate X for each of voter in N_x must be at least as much as their marginal score for x relative to Y. This implies that $S(X \cup \{x\}, \mathbf{v}) - S(X, \mathbf{v}) \geq S(Y \cup \{x\}, \mathbf{v}) - S(Y, \mathbf{v})$ for all complete profiles $\mathbf{v}$. We exploit this full-information submodularity below when analyzing our partial information greedy algorithm.

To develop our greedy approach, we first define the following problem, which we call the *additional alternative problem*. Assume a partial profile $\mathbf{p}$ and a fixed set $\bar{a}$ of $k-1$ alternatives; if one can add a kth alternative to the set, which next option minimizes maximum regret under the Chamberlin-Courant or limited choice model? We define this problem in the obvious way:

$$PMR(a, w, \mathbf{p}|\bar{a}) = PMR(\bar{a} \cup \{w\}, \bar{a} \cup \{a\}, \mathbf{p}) \tag{24}$$

$$MR(a, \mathbf{p}|\bar{a}) = \max_{w \in A} PMR(a, w, \mathbf{p}|\bar{a}) \tag{25}$$

$$MMR(\mathbf{p}|\bar{a}) = \min_{a \in A} MR(a, \mathbf{p}|\bar{a}) \tag{26}$$

$$a^*_{\bar{a},\mathbf{p}} \in \operatorname*{argmin}_{a \in A} MR(a, \mathbf{p}|\bar{a}). \tag{27}$$

Here $PMR(a, w, \mathbf{p}|\bar{a})$ denotes the pairwise max regret of extending slate $\bar{a}$ by adding a rather than w, while $MR(a, \mathbf{p}|\bar{a})$ denotes the regret of extending $\bar{a}$ with a rather than some other alternative in the worst case (over completions of $\mathbf{p}$). Minimax regret and the minimax optimal "extension" of $\bar{a}$, i.e., $a^*_{\bar{a},\mathbf{p}}$, are defined in the obvious way. We note that setting $k = 1$ gives the single-winner robust voting problem addressed earlier.

The additional alternative problem can be solved in polynomial time. We can explicitly compute the pairwise max regret $PMR(a, w, \mathbf{p}|\bar{a})$ of all $m(m-1)/2$ pairs of alternatives (a, w) (where a is a proposed additional option and w is an adversarial witness), using ideas very similar to those developed in Section 4.2 (as we discuss below). Indeed, while we can apply our previous algorithm for finding the pairwise max regret for arbitrary pairs of slates $(\bar{a}, \overline{w})$, the algorithm and analysis of $PMR(a, w, \mathbf{p}|\bar{a})$ that we provide below offers a factor k speedup over the direct application of our earlier algorithm.

With PMR in hand, we can readily determine minimax regret using Eqs. (25) and (26). Hence, we need only show that PMR can be computed in polynomial time. As in Section 4.2, we can compute each voter's i's contribution $PMR(a, w, p_i|\bar{a})$ independently. We again consider two cases.

Case 1, where $Nec_i(a \succ w)$. We have that $PMR(a, w, p_i|\bar{a}) \leq 0$ since adding w to $\bar{a}$ cannot improve i's score any more than adding a. Because $Nec_i(a \succ w)$, if there is some $b \in \bar{a}$ such that $Nec_i(b \succ a)$ or $Inc_i(a, b)$ then b can be ordered over a and $PMR(a, w, p_i|\bar{a}) = 0$. However, if $Nec_i(a \succ b)$ for all $b \in \bar{a}$, regret must be negative. $PMR(a, w, p_i|\bar{a})$ is then maximized (or negative regret minimized) by placing as few alternatives as possible between a and the best element of $\bar{a} \cup \{w\}$. For any b with $Nec_i(a \succ b)$, define

$$T_i(a, b) = \{b' : Nec_i(a \succ b') \text{ and } Nec_i(b' \succ b)\}.$$

Then regret is maximized by ordering the alternatives in $u_i(\bar{a} \cup \{w\})$ such that the element with the fewest possible alternatives between it and a is ranked first. This gives:

$$PMR(a, w, p_i|\bar{a}) = \max_{b \in u_i(\bar{a} \cup \{w\})} -|T_i(a, b)| - 1.$$

Case 2, where $Nec_i(w \succ a)$ or $Inc_i(a, w)$. We have that $PMR(a, w, p_i|\bar{a}) \geq 0$. In this case, if there is some $b \in \bar{a}$ such that $Nec_i(b \succ w)$, then w can never be selected; but since w can be ordered over a, $PMR(a, w, p_i|\bar{a}) = 0$. However, if there is no $b \in \bar{a}$ with $Nec_i(b \succ w)$, then regret is maximized by maximizing the gap between w and the best element of $\bar{a} \cup \{a\}$.

Algorithm 1 greedy algorithm.

$\bar{a} \leftarrow \emptyset$
for $k = 1$ **to** K **do**
  $MMR \leftarrow \infty$
  for $a \in A$ **do**
    $MR \leftarrow -\infty$
    for $w \in A : w \neq a$ **do**
      for $i \in N$ **do**
        if $Nec_i(a \succ w)$ **then**
          $PMR \leftarrow PMR + \max_{b \in u_i(\bar{a} \cup \{w\})} -|T_i(a, b)| - 1$
        else
          $PMR \leftarrow PMR + |B_i'(\bar{a} \cup \{a\}, w)| + 1$
        end if
      end for
      if $PMR > MR$ **then**
        $MR \leftarrow PMR$
      end if
    end for
    if $MR < MMR$ **then**
      $MMR \leftarrow MR$
      $a^* \leftarrow a$
    end if
  end for
  $\bar{a} \leftarrow \bar{a} \cup \{a^*\}$.
end for

In particular, the alternatives in $B_i'(\bar{a} \cup \{a\}, w)$, as defined in Eq. (23), can all be ordered between w and the best such alternative. This gives us:

$$PMR(a, w, p_i|\bar{a}) = |B_i'(\bar{a} \cup \{a\}, w)| + 1.$$

Taken together, this shows:

Theorem 13. *Given a (partial) slate $\bar{a}$, partial vote p_i, and two alternatives $a, w \in A$, pairwise max regret $PMR(a, w, p_i|\bar{a})$ for the additional alternative problem can be computed in polynomial time.*

This gives rise to a very simple *greedy algorithm* for approximating a minimax optimal K-set: starting with the empty slate $\bar{a}_0 = \emptyset$, at each iteration $k \le K$ we add alternative $a_k^* = a^*_{\bar{a}_{k-1},\mathbf{p}}$ to slate $\bar{a}_{k-1}$, i.e., the option with least max regret given the alternatives already on the slate. The method is detailed in Algorithm 1. While this algorithm comes with no strong approximation guarantees—though we provide some weaker guarantees below—we show in Section 7.2 that it appears works well in practice on a small selection of problems.

In terms of run time, case 1 takes $O(mk)$ time (at the kth iteration) and case 2, as discussed previously, takes $O(k^2 + mk) = O(mk)$ time. Computing pairwise max regret for all pairs (a, w), across all agents, and finding the next best alternative $a^*_{\bar{a}_k,\mathbf{p}}$ for each of the K slots on the slate results in a total running time of $O(nm^3K^2)$.

There are two reasons the greedy algorithm is not guaranteed to find the minimax optimal slate. The first is unrelated to preference uncertainty: *even with complete preference information*, the greedy algorithm is unable to ensure we find an optimal K-slate. In other words, it can produce a slate that has positive max regret. However, the greedy algorithm does provide a $1 - \frac{1}{e}$ approximation in the full information setting. It is not hard to see that if we have sufficient information to make the "optimal greedy choice" at each stage, then the regret-based approach will correspond to the exact greedy algorithm described by Lu and Boutilier [62]:

Proposition 14. *If $MR(a_k^*, \mathbf{p}|\bar{a}_{k-1}) = 0$ for all $k \le K$, then the greedy-MMR set $\bar{a}_K$ is identical to the set produced by the (full-information) greedy algorithm given any $\mathbf{v} \in C(\mathbf{p})$.*

Proof. We prove this by induction on the number of greedy-*MMR* iterations k. In the base case, the full information greedy algorithm will select a single alternative, one with the highest score (i.e., the winner under a positional scoring rule). If $MR(a_1^*, \mathbf{p}) = 0$, this implies a_1^* is a necessary winner and hence must have the greatest score, so greedy-*MMR* selects the score optimal alternative.

For the induction, let $\mathbf{v}^*$ be the true but unknown completion of $\mathbf{p}$. For some $k < K$, suppose that the following holds regarding the greedy-*MMR* slate $\bar{a}_k$: each a_i^* added to the greedy-*MMR* slate at iteration i, for $i \le k$, satisfies $a_i^* \in \operatorname{argmax}_{a' \in A} S(\bar{a}_{i-1} \cup \{a'\}, \mathbf{v}^*)$, where $\bar{a}_{i-1} = \{a_1^*, \ldots, a_{i-1}^*\}$. Consider the $k+1$st iteration: at that point, the chosen alternative a_{k+1}^* has conditional max regret zero, i.e., $MR(a_{k+1}^*, \mathbf{p}|\bar{a}_k) = 0$. By definition of conditional max regret, any other alternative $a' \in A$ satisfies $S(\bar{a}_{k-1} \cup \{a'\}, \mathbf{v}) \le S(\bar{a}_k, \mathbf{v})$ for any completion $\mathbf{v} \in C(\mathbf{p})$. In particular, this holds for $\mathbf{v} = \mathbf{v}^*$, which implies, by definition, that a_{k+1}^* is a full-information greedy optimal choice at iteration $k+1$. □

If the last alternative added has non-zero max-regret, we are assured that true minimax regret is also nonzero:

Observation 15. If $MR(a^*_K, \mathbf{p}|\bar{a}_{K-1}) > 0$, then $MMR(\mathbf{p}) > 0$.

Unfortunately, we cannot be sure that if only the last element has zero regret that we have found the greedy-optimal slate. But even if the minimax-optimal alternative a^*_i does not have zero max regret, we can still obtain bounds on the quality of the solution. We can consider each iteration of our greedy algorithm under partial information as *approximating* the corresponding full-information greedy algorithm—it is only able to *approximate the greedy choice* of alternative at each iteration because it is working with the partial profile $\mathbf{p}$ rather than the true, unknown profile $\mathbf{v}$. However, using known results for bounding the quality of *approximate greedy optimization* [47], we can provide a bound on the quality of the slate $\bar{a}^{grd}_K(\mathbf{p})$ produced by the greedy algorithm under partial information (with respect to the *true profile* $\mathbf{v}$) relative to the true optimal slate $\bar{a}^*_K(\mathbf{v})$ given full information. Specifically:

Proposition 16. *Let $\mathbf{p}$ be a (known) partial profile and $\mathbf{v} \in C(\mathbf{p})$ be any (unknown) complete preference profile. Let $\bar{a}^{grd}_K(\mathbf{p}) = \{a^p_1, \ldots, a^p_K\}$ denote the size K slate produced by the regret-based greedy algorithm using partial profile $\mathbf{p}$, where alternative a^p_k is added in the kth iteration, and let $\bar{a}^*_K(\mathbf{v})$ be the optimal K-slate. Let m_k be any lower bound on the marginal value of the kth alternative added to a slate by the* full information *greedy algorithm; that is, $m_k \le S(\bar{a}^f_{k-1} \cup \{a^f_k\}, \mathbf{v}) - S(\bar{a}^f_{k-1}, \mathbf{v})$ where, $\bar{a}^f_{k-1}$ consists of the first $k-1$ greedily selected alternatives using the full profile $\mathbf{v}$. If $MR(a^p_k, \mathbf{p}|\{a^p_1, \ldots a^p_{k-1}\}) \le \frac{\alpha-1}{\alpha} m_k$ for all $k \le K$, for some $\alpha \ge 1$, then*

$$S(\bar{a}^{grd}_K(\mathbf{p}), \mathbf{v}) \ge \left(1 - \frac{1}{e^{1/\alpha}}\right) S(\bar{a}^*_K(\mathbf{v}), \mathbf{v}).$$

In other words, the greedy regret-based algorithm constructs a slate that is within a factor of $1 - \frac{1}{e^{1/\alpha}}$ of the optimal (full-information) slate despite working with incomplete information.

Proof. We prove this bound by relating the marginal improvement offered at the kth iteration of the greedy algorithm when run using full information $\mathbf{v}$ (call this the full-information greedy algorithm) with that when using conditional max regret to add an alternative to the slate (call this the partial-information algorithm). Let $\bar{a}^p_{k-1}$ denote the slate constructed by the partial-information algorithm at iteration $k-1$. Let a^f_k be the alternative that would be added by full-information greedy to slate $\bar{a}^p_{k-1}$ constructed to that point, and let $FI_k = S(\bar{a}^p_{k-1} \cup \{a^f_k\}, \mathbf{v}) - S(\bar{a}^p_{k-1}, \mathbf{v})$ denote its marginal value. Similarly, let a^p_k denote the alternative added by the partial-information algorithm, and let $PI_k = S(\bar{a}^p_{k-1} \cup \{a^p_k\}, \mathbf{v}) - S(\bar{a}^p_{k-1}, \mathbf{v})$ denote its marginal value. Note that we are measuring marginal value relative to the *true* underlying profile $\mathbf{v}$ (notwithstanding the fact that the partial-information algorithm does not have access to this profile). Finally, let $\bar{a}^{grd}_K(\mathbf{p})$ denote the K-slate produced by the partial-information algorithm and $PI = S(\bar{a}^{grd}_K(\mathbf{p}), \mathbf{v})$ denotes its value under $\mathbf{v}$, and let $OPT = S(\bar{a}^*_K(\mathbf{v}), \mathbf{v})$ be the value of the optimal slate $\bar{a}^*_K(\mathbf{v})$.

At stage k, the partial-information algorithm adds an alternative a^p_k that minimizes conditional max regret. By definition of conditional max regret, the marginal value of a^p_k w.r.t. $\bar{a}^p_{k-1}$ must be within an additive factor $MR(a^p_k, \mathbf{p}|\bar{a}_{k-1})$ of the marginal value of a^f_k. Thus we have

$$FI_k \le PI_k + MR(a^p_k, \mathbf{p}|\bar{a}_{k-1}) \tag{28}$$

$$\le PI_k + \frac{\alpha-1}{\alpha} m_k \tag{29}$$

$$\le PI_k + \frac{\alpha-1}{\alpha} FI_k \tag{30}$$

(where the second and third inequalities follow by the statement of the theorem). This implies $\frac{1}{\alpha} FI_k \le PI_k$; i.e., the partial-information greedy algorithm offers an $\frac{1}{\alpha}$-approximation of the full-information algorithm w.r.t. the additional alternative problem.

It can be shown that a greedy algorithm applied to a submodular optimization problem in which the alternative added at each step does not maximize marginal improvement, but does provide a γ-approximation of the maximal marginal improvement, provides a $(1 - \frac{1}{e^\gamma})$ approximation to the submodular optimization problem [47]. Since slate optimization for the full information problem is submodular [62], we have

$$PI \ge \left(1 - \frac{1}{e^{1/\alpha}}\right) OPT. \qquad \square$$

We provide some interpretation of this result and its applicability. The first thing to note is that the result provides a post-hoc (i.e., data- or instance-dependent) bound rather than a prior guarantee on solution quality. Specifically, the quality guarantee requires testing the condition $MR(a_k^p, \mathbf{p} \mid \{a_1^p, \ldots, a_{k-1}^p\}) \leq \frac{\alpha-1}{\alpha} m_k$ with each item added to the slate. This data-dependence is to be expected in the general partial information context we deal with: we would not expect any reasonable bound to apply, say, if we had zero knowledge of voter preferences. Moreover, since we largely use our quality guarantees to drive the decision whether to elicit further preference information, if the conditions of the bound fail, then it suggests we elicit further (see the next section).

We also remark on the condition that we have a lower bound m_k on the marginal improvement of the *full-information* greedy algorithm. In general, such bounds will be problem-dependent and may be hard to obtain. But non-trivial bounds may be viable in many settings. For instance, if we use plurality scoring and have upper and lower bounds on the number of voters in a population that prefer any individual candidate, one can easily derive a lower bound m_k for the kth addition to a slate. Similarly, in marketing, assortment optimization, product line design, etc., preliminary consumer surveys and past purchase behavior may inform such bounds. Finally, if we have distributional information about voter/consumer preferences, we may be able to derive bounds of this type that apply with high probability, and thus derive a probabilistic variant of this result.

Finally, we emphasize that this bound is relative to the quality of the optimal slate that can be obtained under *full information*. Of course, it would be interesting to compare the greedy result with regret of minimax-optimal slate under the same partial information. But the relationship to the quality of the optimal solution is potentially just as valuable a device for driving an elicitation procedure. (Indeed, *MMR* is itself a bound on solution quality relative to the true optimum.)

7. Preference elicitation for multi-winner problems

We now turn our attention to the question of incremental elicitation of voter preferences. When attempting to find an optimal K-slate, with a partial preference profile $\mathbf{p}$, we cannot guarantee that an optimal slate can be obtained (regardless of whether we resort to greedy or exact optimization)—specifically, if $MMR(\mathbf{p}) > 0$, no slate can be guaranteed to be optimal. To improve the quality of the slate, further information must be elicited from one or more voters. Our goal is to reduce *relevant* uncertainty, i.e., find those queries that have the greatest potential to reduce minimax regret. To do this we adapt the principles of the single-winner *current solution strategy (CSS)*, introduced in Section 5, to work with slates. We first define CSS in this new context, then provide empirical evaluation of CSS, both in terms of the number of queries required to reach solutions with low minimax regret, and its running time.

7.1. Current solution strategy for slate selection

We focus here on pairwise comparison queries, in which some voter i is asked whether $a \succ_i a'$; but the basic principles can be applied to other forms of queries (e.g., top-t queries, as discussed in Section 5). The use of CSS differs depending on whether we are using the greedy heuristic or optimal *MMR* computation to determine our K-slates. We present our approach in the context of greedy-*MMR* computation since it is the more practical method for problems involving slates of reasonable size. The general principles can be readily adapted to optimal *MMR* computation as well.

Our elicitation scheme works by using the greedy algorithm to compute an (approximately) minimax optimal slate $\bar{a}^*_{\mathbf{p}} = \langle a_1^*, \ldots, a_K^* \rangle$ given the current partial profile $\mathbf{p}$. If $MR(a_K^*, \mathbf{p}|\bar{a}^*_{K-1}) = 0$, we treat this as an (approximately) minimax optimal slate and stop. Otherwise, we know that $MMR(\mathbf{p}) > 0$, so we use CSS to select a voter i and a pairwise comparison query $a \succ_i a'$ with the greatest potential to reduce $MR(a_K^*, \mathbf{p}|\bar{a}^*_{K-1})$. Let a_K^* be the last alternative added to the slate and let w_K be the witness alternative (i.e., where $MR(a_K^*, \mathbf{p}|\bar{a}^*_{K-1}) = PMR(a_K^*, w_K, \mathbf{p}|\bar{a}^*_{K-1})$. CSS identifies the appropriate query (and its potential) for a particular voter i based on several specific cases.

Case 1, where $Nec_i(a_K^* \succ w_K)$. Then voter i's contribution PMR_i to $PMR(a_K^*, w_K, \mathbf{p}|\bar{a}^*_{K-1})$ must be $PMR_i \leq 0$. If $PMR_i = 0$, then either: (i) a_K^* is dominated in i's partial ranking p_i by some $a_j \in \bar{a}^*_{K-1}$, or (ii) a_K^* is not dominated by any such a_j. In case (i), no query can reduce $MR(a_K^*, \mathbf{p}|\bar{a}^*_{K-1})$ since voter i would never select either of a_K^* or w_K given the rest of the slate $\bar{a}^*_{K-1}$, so *no query* is asked of i. In case (ii), the adversary can set $PMR_i = 0$ by ordering some alternative $a_j \in \bar{a}^*_{K-1}$ over a_K^* (if no such alternative were possible, PMR_i would have to be negative). In this case, any query that prevents a_j from being orderable above a_k can reduce PMR_i (by making it negative). Specifically, any query of the form $b \succ_i c$ for $b \in a_K^* \cup \{a : Nec_i(a_K^* \succ a)\}$ and $c \in a_j \cup \{a : Nec_i(a \succ a_j)\}$ will suffice. Since the degree of PMR_i is determined by the relationship of a_K^* to w_K and *not the gap* with a_j, we choose query $a_K^* \succ_i a_j$ since it is implied by any other.

If $PMR_i < 0$, then (iii) a_K^* dominates each $a_j \in \bar{a}^*_{K-1}$ as well as w_K. Queries that can extend the advantage over the best alternative of $w^* \in u_i(\{w_K\} \cup \bar{a}^*_{K-1})$ (i.e., make PMR_i even smaller) take two forms: learning that $Nec_i(a_K^* \succ c)$ for some ancestor c of w^* (since we do not know w^*, we choose an arbitrary alternative in $u_i(\{w_K\} \cup \bar{a}^*_{K-1})$); or learning that $Nec_i(d \succ w^*)$ for some descendant d of a_K^*. The query with the greatest potential is that with the largest number of descendants (respectively, ancestors) lying between it and c (respectively, d) in p_i.

Case 2, where $Nec_i(w_K \succ a_K^*)$ or $Inc_i(a_K^*, w_K)$. Then voter i's contribution PMR_i to $PMR(a_K^*, w_K, \mathbf{p}|\bar{a}^*_{K-1})$ must be $PMR_i \geq 0$. (i) If $PMR_i = 0$, then we must have w_K dominated by some $a_j \in \bar{a}^*_{K-1}$. In this case we ask no query. (ii) If $PMR_i > 0$, then w_K is not dominated by any $a_j \in \bar{a}^*_K$ (i.e., by none of the K alternatives). Then regret can be reduced only by asking a query that removes elements from the set $B'_i(\bar{a}^*_K, w_K)$ by either: (a) placing w_K (or one of its ancestors) below some a_j (it is only necessary to consider those in $u_i(\bar{a}^*_K)$); (b) placing some $a_j \in u_i(\bar{a}^*_K)$ (or one of its descendants) above w_K; or (c) placing some element that is incomparable to both a^*_K and w_K either above w_K (hence placing its ancestors above as well) or below a^*_K (hence placing its descendants below as well). In the case that w_K dominates all elements of $\bar{a}^*_K$, one can ask queries that either (a) move a descendant d of w_K, where d is not an ancestor of some $a_j \in \bar{a}^*_K$, below such an a_j; or (b) move an ancestor c of some a_j, where c is not a descendant of w_K, above w_K. The potential of a query to reduce PMR_i is measured by the number of elements it removes from the set $B'_i(\bar{a}^*_K, w_K)$.

We note that if $PMR(a^*_K, w_K, \mathbf{p}|\bar{a}^*_{K-1}) > 0$, then one of the query generating cases above must hold for at least one voter. As a consequence, CSS cannot "stall" as long as the last alternative added to the slate has non-zero max regret.

7.2. Empirical evaluation

We now describe experiments designed to test the ability of our greedy slate optimization method, when coupled with the CSS elicitation strategy, to find good (or even optimal) slates of alternatives with few voter queries. We evaluate the approach on two real datasets as well as on more systematically generated random data. As in single-winner problems, we use the *Sushi* data set and the *Irish* dataset, and draw profiles of 100 random users/voters from these sets as described in Section 5.3. In the Sushi experiments we set the slate size to be $K = 3$, while in the Irish experiments we set $K = 4$. We also test CSS by generating random profiles of 100 voter rankings over 10 alternatives, with voter rankings drawn i.i.d. from a Mallows distribution, again using the same methodology as in Section 5.3. In the *Mallows* experiments, we set $K = 3$ and analyze elicitation performance as we vary the dispersion parameter ϕ.

Each experimental instance consists of a full profile. We start each run with no voter preference information, then, using CSS to generate queries, we elicit pairwise comparisons from voters (who respond accurately based on their underlying preferences). After each query/round, we use the greedy algorithm to compute an approximately optimal K-slate. Elicitation terminates once the conditional max regret (CondMR) of the Kth alternative added to the slate, $MR(a^*_K, \mathbf{p}|\bar{a}^*_{K-1})$, is zero. (Note that after each elicitation round a "fresh" slate is constructed; elicitation is terminated at that round if the Kth alternative added to the slate *at that round* has zero conditional max regret.) We also compare this to the use of exact minimax regret computation at each round of elicitation to determine the truly optimal K-slate.

Results in Figs. 6(a) and (b) show performance for the *Sushi* and *Irish* data. All results are averaged over 20 randomly drawn profiles, and in each plot we show the following metrics for the slate produced by the greedy algorithm after each query:

- its conditional max regret (CondMR), i.e., that of the last element added to the slate;
- its true max regret;
- and one sample standard deviation above and below its regret values (dotted lines).

We also show the performance of CSS when exact *MMR* is computed and the optimal slate is generated at each iteration. Finally, we compare CSS with a baseline random strategy (Rand) that randomly picks a voter and pairwise comparison query (ensuring this response to this query is not implied by that voter's partial ranking, i.e., it is not contained in the transitive closure of previous responses), using Greedy to compute the slate at each round and measuring its max regret. Again, a range of one standard deviation is shown.

These plots indicate that CSS works very well. It finds a slate with zero max regret "per voter" with only about 20 queries per user in Sushi (respectively, 15 in Irish). CSS also reaches near-zero regret in about 10 queries (respectively, 8) per user; thus, its *anytime profile* is very encouraging for settings where approximately optimal solutions are permissible, especially if approximation provides a significant reduction in elicitation burden. Note that the true regret may be zero even if max regret is not. We contrast the number of queries needed by CSS with the demands of complete sorting to provide a full ranking, which requires $O(m \log(m))$ pairwise comparisons using methods with good average case performance, or equivalently 34 (respectively, 29) queries. Random requires 25 (respectively, 22) queries per user to reach zero regret, and has a much worse anytime profile.

The greedy algorithm itself works extremely well: it almost always finds the minimax optimal slate—the *MR* Greedy and *MMR* curves coincide almost exactly—and in the rare cases that it does not, Greedy *MR* is very close to true *MMR*. MR may not decrease monotonically, as preference updates may "mislead" Greedy into choosing an inferior slate (by contrast, true *MMR* is non-increasing). CondMR is also a good proxy for true max regret: in *Sushi*, the per-voter difference is at most 0.41 and in *Irish* at most 0.24. Thus, CondMR—which can be computed efficiently—is an excellent surrogate for *MR*—which is NP-hard—as a quality measure and a stopping criterion for elicitation.

Mallows results in Fig. 6 (c) show how the same quantities change as a function of the total number of queries, for different dispersion values ϕ. The results show that, unsurprisingly, more concentrated preference distributions (smaller ϕ) require fewer queries to find good slates. This is consistent with results for the single winner voting (see Section 5.3).

Table 2 shows wall clock runtimes for Greedy with different values of m (alternatives) and K (slate size) on a 3.0 GHz Intel Xeon processor. Results are averaged over "complete" CSS elicitation runs (i.e., over all optimizations solved until elicitation reaches a slate whose CondMR is zero), on random profiles of $n = 40$ voters drawn from a Mallows distribution with $\phi = .7$. Average runtime increases significantly with the number of candidates m, but less dramatically with K. This is

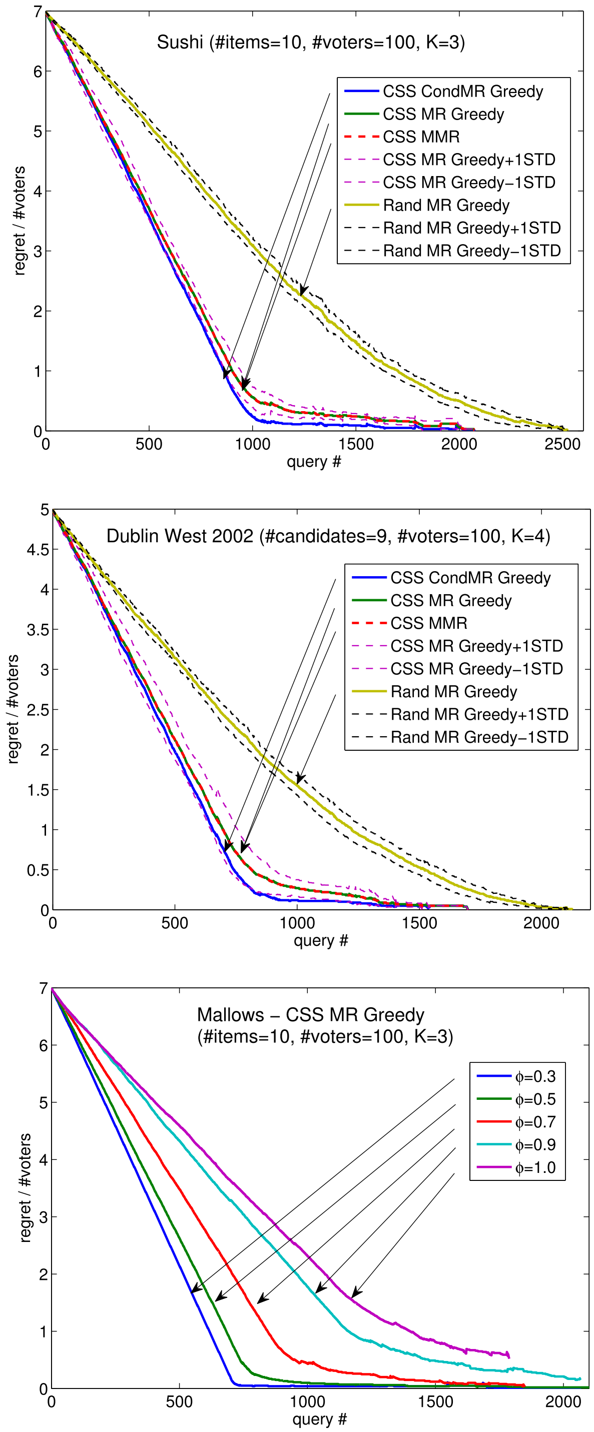

Fig. 6. Performance of Greedy/CSS on (a) *Sushi* (20 trials), (b) *Irish* (20 trials) and (c) *Mallows* datasets (20 trials).

Table 2
Average Greedy runtime (sec.), on random Mallows profiles.

m	$K=2$	$K=3$	$K=4$	$K=6$	$K=8$
10	0.015	0.020	0.023	0.028	0.033
20	0.105	0.152	0.194	0.275	0.345
30	0.342	0.508	0.642	0.987	1.282
50	1.577	2.042	2.247	4.439	6.344

consistent with the "quadratic in K" and "cubic in m" computational analysis described above. Still, Greedy is very practical, taking only 6.3 s to find optimal slates for $m = 50$ candidates and slates of size $K = 8$.

8. Concluding remarks

We have proposed the use of minimax regret as a means of robust group decision support for both single-winner determination and a certain form of the slate selection problem. This regret criterion allows us to analyze the quality of the recommended outcome by bounding the loss in "social welfare," as indicated by the profile score of the selected alternatives, with respect to the optimal outcome. Consequently, this approach supports the informational approximation of voting rules, as well as guiding the process of incremental elicitation of voter preferences.

We have demonstrated the tractability of regret computation for several important single-winner voting rules, and showed the power of regret-based elicitation on both real-world and synthetic data sets. Our empirical results for single-winner regret-based elicitation show that one can determine both approximate and exact winners using only a small fraction of (pairwise) voter preferences. In multi-winner settings, we extended the approach to deal with Chamberlin-Courant style schemes. We developed algorithms for both the exact and greedy optimization of slates with respect to minimax regret. The greedy method is especially practical from a computational perspective. We provided some theoretical bounds on the performance of the greedy method, but more importantly demonstrated that it often outputs slates that are either minimax optimal or very close to optimal. We also adapted the CSS elicitation heuristic and showed that, when coupled with the greedy slate algorithm (even when using conditional max regret as surrogate for max regret), it finds very good slates while asking for relatively little preference information. As with single-winner CSS, it has a desirable anytime profile.

While our results suggest that incremental elicitation is viable in many practical domains, a number of interesting avenues for future research remain. Apart from developing computational and elicitation schemes for additional single- and multi-winner voting rules, one important direction is to develop approaches well-suited for multi-attribute domains, where alternatives and/or voters are specified using particular instantiations of attributes, and preferences are represented as compact functions of these attributes. Such models are particularly relevant in recommender systems and product configuration.

Another direction for future work is the deeper empirical analysis of both our regret-based robust optimization techniques and the CSS elicitation method on a wider range of data sets, such as those contained in the PrefLib repository [69].

While our elicitation techniques are distribution-free, future directions include tailoring elicitation to the demands of particular population preference distributions (e.g., Mallows [68], Plackett-Luce [66,76], etc.). Such a probabilistic framework would allow for more subtle analysis of elicitation performance and new elicitation heuristics. The models could be purely (Bayesian) decision-theoretic; but more interesting analyses would mix probabilistic and regret-based reasoning (e.g., the expected number of queries needed to determine a winner or to reduce minimax regret to some acceptable threshold). This is an important step in making incremental vote elicitation practical in real-world settings. Some preliminary work in this direction couples regret-based decision making with query policies influenced by prior distributions over voter preferences within a Bayesian PAC-style framework [64]. Recently, several other probabilistic and learning-based models for social choice with partial information have been developed as well [40,45,74].

At the same time, minimizing the number of rounds of elicitation by asking many (or all) voters to provide some preference information simultaneously, rather than in sequence, and allowing voters to provide greater amounts of information at each round, can provide important practical advantages. As discussed in Sections 2.4 and 5.3, the scheme of Kalech et al. [53] does just this in a heuristic way, providing solutions with far fewer rounds of elicitation than our scheme. However, they do not analyze the tradeoff between rounds of elicitation, the amount of information ultimately elicited, and the quality of the final solution. A framework for such an analysis is developed by Lu and Boutilier [64].

Finally, there are interesting questions connecting elicitation to vote manipulation, where the elicitation process may reveal preference information that a manipulating coalition can exploit. Neither our optimization techniques nor our elicitation methods account for the strategic nature of social interactions. Strategy-proof (non-regret-based) elicitation has been considered for some voting rules (e.g., [33,59]). While the strategic nature of regret-based elicitation has been studied in the context of mechanism design [50–52], exploring the impact of regret-based elicitation on opportunities for vote manipulation remains of significant interest.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Hassene Aissi, Cristina Bazgan, Daniel Vanderpooten, Min-max and min-max regret versions of combinatorial optimization problems: a survey, Eur. J. Oper. Res. 197 (2) (2009) 427–438.
[2] Fuad Aleskerov, Vyacheslav V. Chistyakov, Valery Kalyagin, The threshold aggregation, Econ. Lett. 107 (2) (2010) 261–262.
[3] Kenneth J. Arrow, A difficulty in the concept of social welfare, J. Polit. Econ. 58 (August 1950) 328–346.
[4] Igor Averbakh, Minmax regret solutions for minimax optimization problems with uncertainty, Oper. Res. Lett. 27 (2000) 57–65.

[5] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, Toby Walsh, Justified representation in approval-based committee voting, Soc. Choice Welf. 48 (2) (2017) 461–485.
[6] Haris Aziz, Piotr Faliszewski, Bernard Grofman, Arkadii Slinko, Nimrod Talmon, Egalitarian committee scoring rules, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-18, Stockholm, 2018, pp. 56–62.
[7] Haris Aziz, Serge Gaspers, Joachim Gudmundsson, Simon Mackenzie, Nicholas Mattei, Toby Walsh, Computational aspects of multi-winner approval voting, in: Proceedings of the Fourteenth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-15, Istanbul, 2015, pp. 107–115.
[8] Yoram Bachrach, Nadja Betzler, Piotr Faliszewski, Probabilistic possible winner determination, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI-10, Atlanta, GA, 2010, pp. 697–702.
[9] Dorothea Baumeister, Piotr Faliszewski, Jérôme Lang, Jörg Rothe, Campaigns for lazy voters: truncated ballots, in: Proceedings of the Eleventh International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-12, Valencia, Spain, 2012, pp. 577–584.
[10] Dorothea Baumeister, Jörg Rothe, Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules, in: Proceedings of the Nineteenth European Conference on Artificial Intelligence, ECAI-10, Lisbon, 2010, pp. 1019–1020.
[11] Nawal Benabbou, Serena Di Sabatino Di Diodoro, Patrice Perny, Paolo Viappiani, Incremental preference elicitation in multi-attribute domains for choice and ranking with the Borda count, in: International Conference on Scalable Uncertainty Management, Nice, France, Springer, 2016, pp. 81–95.
[12] Nadja Betzler, Britta Dorn, Towards a dichotomy for the possible winner problem in elections based on scoring rules, J. Comput. Syst. Sci. 76 (8) (2010) 812–836.
[13] Nadja Betzler, Arkadii Slinko, Johannes Uhlmann, On the computation of fully proportional representation, J. Artif. Intell. Res. 47 (2013) 475–519.
[14] Craig Boutilier, Computational decision support: regret-based models for optimization and preference elicitation, in: P.H. Crowley, T.R. Zentall (Eds.), Comparative Decision Making: Analysis and Support Across Disciplines and Applications, Oxford University Press, Oxford, 2013, pp. 423–453.
[15] Craig Boutilier, Fahiem Bacchus, Ronen I. Brafman UCP-Networks, A directed graphical representation of conditional utilities, in: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, UAI-01, Seattle, 2001, pp. 56–64.
[16] Craig Boutilier, Relu Patrascu, Pascal Poupart, Dale Schuurmans, Constraint-based optimization and utility elicitation using the minimax decision criterion, Artif. Intell. 170 (8–9) (2006) 686–713.
[17] Craig Boutilier, Kevin Regan, Paolo Viappiani, Online feature elicitation in interactive optimization, in: Proceedings of the Twenty-Sixth International Conference on Machine Learning, ICML-09, Montreal, 2009, pp. 73–80.
[18] Craig Boutilier, Jeffrey Rosenschein, Incomplete information and communication in voting (Chapter 10), in: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A.D. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, 2015.
[19] Craig Boutilier, Tuomas Sandholm, Rob Shields, Eliciting bid taker non-price preferences in (combinatorial) auctions, in: Proceedings of the Nineteenth National Conference on Artificial Intelligence, AAAI-04, San Jose, CA, 2004, pp. 204–211.
[20] Steven Brams, Peter Fishburn, Approval voting, Am. Polit. Sci. Rev. 72 (3) (1978) 831–847.
[21] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, Ariel Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, Cambridge, 2016.
[22] Darius Braziunas, Craig Boutilier, Minimax regret-based elicitation of generalized additive utilities, in: Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, UAI-07, Vancouver, 2007, pp. 25–32.
[23] Darius Braziunas, Craig Boutilier, Assessing regret-based preference elicitation with the UTPREF recommendation system, in: Proceedings of the Eleventh ACM Conference on Electronic Commerce, EC'10, Cambridge, MA, 2010, pp. 219–228.
[24] Ioannis Caragiannis, Dimitris Kalaitzis, Evangelos Markakis, Approximation algorithms and mechanism design for minimax approval voting, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI-10, Atlanta, 2010, pp. 737–742.
[25] Ioannis Caragiannis, Ariel D. Procaccia, Nisarg Shah, When do noisy votes reveal the truth?, in: Proceedings of the Fourteenth ACM Conference on Electronic Commerce, EC'13, Philadelphia, 2013, pp. 143–160.
[26] John R. Chamberlin, Paul N. Courant, Representative deliberations and representative decisions: proportional representation and the Borda rule, Am. Polit. Sci. Rev. 77 (3) (1983) 718–733.
[27] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, Nicolas Maudet, A short introduction to computational social choice, in: Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM-07, Harrachov, Czech Republic, 2007, pp. 51–69.
[28] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, Nicolas Maudet, Preference handling in combinatorial domains: from AI to social choice, AIM Mag. 29 (4) (2008) 37.
[29] Reuven Cohen, Liran Katzir, The generalized maximum coverage problem, Inf. Process. Lett. 108 (1) (2008) 15–22.
[30] Ronan Congar, Vincent Merlin, A characterization of the maximin rule in the context of voting, Theory Decis. 72 (1) (January 2012) 131–147.
[31] Vincent Conitzer, Eliciting single-peaked preferences using comparison queries, J. Artif. Intell. Res. 35 (2009) 161–191.
[32] Vincent Conitzer, Making decisions based on the preferences of multiple agents, Commun. ACM 53 (3) (2010) 84–94.
[33] Vincent Conitzer, Tuomas Sandholm, Vote elicitation: complexity and strategy-proofness, in: Proceedings of the Eighteenth National Conference on Artificial Intelligence, AAAI-02, Edmonton, 2002, pp. 392–397.
[34] Vincent Conitzer, Tuomas Sandholm, Communication complexity of common voting rules, in: Proceedings of the Sixth ACM Conference on Electronic Commerce, EC'05, Vancouver, 2005, pp. 78–87.
[35] Lihi Naamani Dery, Meir Kalech, Lior Rokach, Bracha Shapira, Reaching a joint decision with minimal elicitation of voter preferences, Inf. Sci. 278 (2014) 466–487.
[36] Palash Dey, Arnab Bhattacharyya, Sample complexity for winner prediction in elections, in: Proceedings of the Fourteenth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-15, Istanbul, 2015, pp. 1421–1430.
[37] Palash Dey, Neeldhara Misra, Elicitation for preferences single peaked on trees, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-16, New York, 2016, pp. 215–221.
[38] Palash Dey, Neeldhara Misra, Preference elicitation for single crossing domain, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-16, New York, 2016, pp. 222–228.
[39] Ning Ding, Fangzhen Lin, Voting with partial information: what questions to ask?, in: Proceedings of the Twelfth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-13, St. Paul, MN, 2013, pp. 1237–1238.
[40] John A. Doucette, Kate Larson, Robin Cohen, Conventional machine learning for social choice, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI-15, Austin, TX, 2015, pp. 858–864.
[41] Joanna Drummond, Craig Boutilier, Elicitation and approximately stable matching with partial preferences, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI-13, Beijing, 2013, pp. 97–105.
[42] Cynthia Dwork, Ravi Kumar, Moni Naor, D. Sivakumar, Rank aggregation methods for the web, in: Proceedings of the Tenth International World Wide Web Conference, WWW-01, Hong Kong, ACM, 2001, pp. 613–622.
[43] Edith Elkind, Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, Properties of multiwinner voting rules, Soc. Choice Welf. 48 (3) (2017) 599–632.
[44] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, Nimrod Talmon, Multiwinner voting: a new challenge for social choice theory (Chapter 2), in: Endriss Ulle (Ed.), Trends in Computational Social Choice, AI Access, 2017, pp. 27–47.

[45] Yuval Filmus, Joel Oren, Efficient voting via the top-k elicitation scheme: a probabilistic approach, in: Proceedings of the Fifteenth ACM Conference on Electronic Commerce, EC'14, Palo Alto, 2014, pp. 295–312.
[46] Wulf Gaertner, A Primer in Social Choice Theory. LSE Perspectives in Economic Analysis, Oxford University Press, August 2006.
[47] Pranava R. Goundan, Andreas S. Schulz, Revisiting the greedy approach to submodular set function maximization, Optim. Online (2007).
[48] Mohammad Taghi Hajiaghayi, Mohammad Mahdian, Vahab S. Mirrokni, The facility location problem with general cost functions, Networks 42 (1) (2003) 42–47.
[49] Noam Hazon, Yonatan Aumann, Sarit Kraus, Michael Wooldridge, On the evaluation of election outcomes under uncertainty, Artif. Intell. 189 (2012) 1–18.
[50] Nathanaël Hyafil, Craig Boutilier, Regret-based incremental partial revelation mechanisms, in: Proceedings of the Twenty-First National Conference on Artificial Intelligence, AAAI-06, Boston, 2006, pp. 672–678.
[51] Nathanaël Hyafil, Craig Boutilier, Mechanism design with partial revelation, in: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, IJCAI-07, Hyderabad, India, 2007, pp. 1333–1340.
[52] Nathanaël Hyafil, Craig Boutilier, Partial revelation automated mechanism design, in: Proceedings of the Twenty-Second National Conference on Artificial Intelligence, AAAI-07, Vancouver, 2007, pp. 72–78.
[53] Meir Kalech, Sarit Kraus, Gal A. Kaminka, Claudia V. Goldman, Practical voting rules with partial information, J. Auton. Agents Multi-Agent Syst. 22 (1) (2011) 151–182.
[54] Toshihiro Kamishima, Hideto Kazawa, Shotaro Akaho, Supervised ordering: an empirical survey, in: IEEE International Conference on Data Mining, ICDM-05, Houston, TX, 2005, pp. 673–676.
[55] Jon Kleinberg, Christos Papadimitriou, Prabhakar Raghavan, Segmentation problems, J. ACM 51 (2004) 263–280.
[56] Kathrin Konczak, Jérôme Lang, Voting procedures with incomplete preferences, in: IJCAI-05 Workshop on Advances in Preference Handling, Edinburgh, 2005, pp. 124–129.
[57] Kouvelis Panos, Gang Yu, Robust Discrete Optimization and Its Applications, Kluwer, Dordrecht, 1997.
[58] Jérôme Lang, Maria Silvia Pini, Francesca Rossi, Domenico Salvagnin, Kristen Brent Venable, Toby Walsh, Winner determination in voting trees with incomplete preferences and weighted votes, Auton. Agents Multi-Agent Syst. 25 (1) (2012) 130–157.
[59] David Timothy Lee, Efficient, private, and ε-strategyproof elicitation of tournament voting rules, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-15, Buenos Aires, 2015, pp. 2016–2032.
[60] David Timothy Lee, Ashish Goel, Tanja Aitamurto, Helene Landemore, Crowdsourcing for participatory democracies: efficient elicitation of social choice functions, in: Second AAAI Conference on Human Computation and Crowdsourcing, Pittsburgh, PA, 2014, pp. 133–142.
[61] Lev Omer, Jeffrey S. Rosenschein, Convergence of iterative voting, in: Proceedings of the Eleventh International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-12, Valencia, Spain, 2012, pp. 611–618.
[62] Tyler Lu, Craig Boutilier, Budgeted social choice: from consensus to personalized decision making, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI-11, Barcelona, 2011, pp. 280–286.
[63] Tyler Lu, Craig Boutilier, Robust approximation and incremental elicitation in voting protocols, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI-11, Barcelona, 2011, pp. 287–293.
[64] Tyler Lu, Craig Boutilier, Vote elicitation with probabilistic preference models: empirical estimation and cost tradeoffs, in: Proceedings of the Second International Conference on Algorithmic Decision Theory, ADT-11, Piscataway, NJ, 2011, pp. 135–149.
[65] Tyler Lu, Craig Boutilier, Multi-winner social choice with incomplete preferences, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI-13, Beijing, 2013, pp. 263–270.
[66] R. Duncan Luce, Individual Choice Behavior: A Theoretical Analysis, Wiley, 1959.
[67] Brendan Lucier, Joel Oren, Online (budgeted) social choice, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI-14, Québec City, 2014, pp. 1456–1462.
[68] Colin L. Mallows, Non-null ranking models, Biometrika 44 (1957) 114–130.
[69] Nicholas Mattei, Toby Walsh Preflib, A library for preferences, in: Proceedings of the Third International Conference on Algorithmic Decision Theory (ADT-13), Bruxelles, Belgium, 2013, pp. 259–270, see http://www.preflib.org.
[70] Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein, Nicholas R. Jennings, Convergence to equilibria in plurality voting, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI-10, Atlanta, GA, 2010, pp. 823–828.
[71] Reshef Meir, Ariel D. Procaccia, Jeffrey S. Rosenschein, Aviv Zohar, Complexity of strategic behavior in multi-winner elections, J. Artif. Intell. Res. 33 (2008) 149–178.
[72] Burt L. Monroe, Fully proportional representation, Am. Polit. Sci. Rev. 89 (4) (1995) 925–940.
[73] Hannu Nurmi, Voting Procedures Under Uncertainty, Springer-Verlag, Berlin, 2002.
[74] Joel Oren, Yuval Filmus, Craig Boutilier, Efficient vote elicitation under candidate uncertainty, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI-13, Beijing, 2013, pp. 309–316.
[75] Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, Toby Walsh, Aggregating partially ordered preferences, J. Log. Comput. 19 (2009) 475–502.
[76] R.L. Plackett, The analysis of permutations, Appl. Stat. 24 (1975) 193–202.
[77] Richard F. Potthoff, Steven J. Brams, Proportional representation: broadening the options, J. Theor. Polit. 10 (2) (1998) 147–178.
[78] Ariel D. Procaccia, Jeffrey S. Rosenschein, Aviv Zohar, On the complexity of achieving proportional representation, Soc. Choice Welf. 30 (2008) 353–362.
[79] Kevin Regan, Craig Boutilier, Regret-based reward elicitation for Markov decision processes, in: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI-09, Montreal, 2009, pp. 451–454.
[80] Kevin Regan, Craig Boutilier, Eliciting additive reward functions for Markov decision processes, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI-11, Barcelona, 2011, pp. 2159–2164.
[81] Michel Regenwetter, Bernard Grofman, A.A.J. Marley, Ilia Tsetlin, Behavioral Social Choice: Probabilistic Models, Statistical Inference, and Applications, Cambridge University Press, Cambridge, 2006.
[82] Reyhaneh Reyhani, Mark Wilson, Best-reply dynamics for scoring rules, in: 20th European Conference on Artificial Intelligence, ECAI-12, Montpellier, France, 2012, pp. 672–677.
[83] Magnus Roos, Jörg Rothe, Complexity of social welfare optimization in multiagent resource allocation, in: Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-10, Toronto, 2010, pp. 641–648.
[84] Ahti Salo, Raimo P. Hämäläinen, Preference ratios in multiattribute evaluation (PRIME)—elicitation and decision procedures under incomplete information, IEEE Trans. Syst. Man Cybern. 31 (6) (2001) 533–545.
[85] Piotr Skowron, Piotr Faliszewski, Chamberlin-Courant rule with approval ballots: approximating the MaxCover problem with bounded frequencies in FPT time, J. Artif. Intell. Res. 60 (1) (2017) 687–716, 9.
[86] Piotr Skowron, Piotr Faliszewski, Jérôme Lang, Finding a collective set of items: from proportional multirepresentation to group recommendation, Artif. Intell. 241 (2016) 191–216.
[87] Piotr Skowron, Piotr Faliszewski, Arkadii Slinko, Achieving proportional representation is easy in practice, in: Proceedings of the Twelfth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-13, St. Paul, MN, 2013, pp. 399–406.

[88] Piotr Krzysztof Skowron, Piotr Faliszewski, Arkadii Slinko, Fully proportional representation as resource allocation: Approximability results, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI-13, Beijing, 2013, pp. 353–359.
[89] Warren Smith, Range voting, http://www.math.temple.edu/~wds/homepage/rangevote.pdf, 2000.
[90] Hossein Azari Soufiani, David C. Parkes, Lirong Xia, Preference elicitation for general random utility models, in: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI-13, Bellevue, WA, 2013, pp. 596–605.
[91] Olivier Toubia, John R. Hauser, Duncan I. Simester, Polyhedral methods for adaptive choice-based conjoint analysis, J. Mark. Res. 41 (1) (2004) 116–131.
[92] Paolo Viappiani, Craig Boutilier, Regret-based optimal recommendation sets in conversational recommender systems, in: Proceedings of the 3rd ACM Conference on Recommender Systems, RecSys09, New York, 2009, pp. 101–108.
[93] Toby Walsh, Uncertainty in preference elicitation and aggregation, in: Proceedings of the Twenty-Second National Conference on Artificial Intelligence, AAAI-07, Vancouver, 2007, pp. 672–678.
[94] Toby Walsh, Complexity of terminating preference elicitation, in: Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-08, Estoril, Portugal, 2008, pp. 967–974.
[95] Tianhan Wang, Craig Boutilier, Incremental utility elicitation with the minimax regret decision criterion, in: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI-03, Acapulco, 2003, pp. 309–316.
[96] Lirong Xia, Vincent Conitzer, Determining possible and necessary winners under common voting rules given partial orders, J. Artif. Intell. Res. 41 (2011) 25–67.
[97] Zhibing Zhao, Haoming Li, Junming Wang, Jeffrey Kephart, Nicholas Mattei, Hui Su, Lirong Xia, A cost-effective framework for preference elicitation and aggregation, preprint, arXiv:1805.05287, 2018.