Computational Aspects of Approval Voting¹

Dorothea Baumeister¹

Gábor Erdélyi¹

Edith Hemaspaandra²

Lane A. Hemaspaandra³

Jörg Rothe¹

Univ. of Rochester Comp. Sci. Dept. Technical Report TR-2009-944

May 5, 2009

¹Supported in part by DFG grants RO 1202/11-1 and RO 1202/12-1, NSF grants CCF-0426761 and IIS-0713061, the Alexander von Humboldt Foundation's TransCoop program, the European Science Foundation's EUROCORES program LogICCC, and two Friedrich Wilhelm Bessel Research Awards.

Abstract

This paper is concerned with the computational aspects of approval voting and some of its variants, with a particular focus on the complexity of problems that model various ways of tampering with the outcome of an election: manipulation, control, and bribery. For example, in control settings, the election's chair seeks to alter the outcome of an election via control actions such as adding/deleting/partitioning either candidates or voters. In particular, sincere-strategy preference-based approval voting (SP-AV), a variant of approval voting proposed by Brams and Sanver [BS06], is computationally resistant to 19 of the 22 common types of control. Thus, among those natural voting systems for which winner determination is easy, SP-AV is the system currently known to display the broadest resistance to control. We also present the known complexity results for various types of bribery. Finally, we study local search heuristics for minimax approval voting, a variant of approval voting proposed by Brams, Kilgour, and Sanver [BKS04] (see also [BKS07a, BKS07b]) for the purpose of electing a committee of fixed size.

¹URLs: ccc.cs.uni-duesseldorf.de/~baumeister, [~erdelyi](http://ccc.cs.uni-duesseldorf.de/~erdelyi), [~rothe](http://ccc.cs.uni-duesseldorf.de/~rothe). Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany. Work done in part while the second author was visiting Universität Trier. Corresponding Author: J. Rothe.

²URL: www.cs.rit.edu/~eh. Department of Computer Science, Rochester Institute of Technology, Rochester, NY 14623, USA.

³URL: www.cs.rochester.edu/u/lane. Department of Computer Science, University of Rochester, Rochester, NY 14627, USA.

1 Introduction

“Yes, we can!”—Barack Obama’s campaign slogan inspired enough of his supporters to go to the polls and give him their “yes” votes that he won the 2008 U.S. presidential election. And this happened notwithstanding the fact that many other voters said “No” when pollsters asked if they viewed Barack Obama as qualified for the office. “Yes” and “no” are perhaps the most basic ways for us, as voters, to express our preferences about candidates, and “yes” and “no” are what approval voting is all about.

In approval voting, every voter either approves or disapproves of each candidate. At the end of the day, all approvals are counted and whoever is approved of by the most voters wins. Since Brams and Fishburn [BF78] proposed this system three decades ago, it has been studied intensely in social choice theory (see, e.g., [Bra80,BF81,BF83,BF02,BKS04,BF05,DSL06,BS06,KBS06,BKS07a,BS09]), it has been adopted by numerous scientific and engineering societies (such as the IEEE),¹ and it has even been dubbed “the electoral reform of the twentieth century” by its proponents (see Dutta, De Sinopoli, and Laslier [DSL06]). This paper focuses on the computational aspects of approval voting.

Why should one bother to study the computational aspects of approval voting? Isn’t this just a matter of summing up the approvals each candidate receives and comparing the resulting approval scores? Sure enough, adding and comparing are quite easy computational tasks, and this is one important point in favor of approval voting: It is easy to determine the winners of an approval election. However, “computational aspects” here is meant in a much broader and deeper sense than with regard to mere winner determination. In particular, for approval voting (AV, for short) and one of its variants (SP-AV), this paper discusses computational issues that model attempts to tamper with the outcome of an election in various ways, and we will pay particular attention to the complexity results known for these computational problems.

For example, the issue of voters tampering with election outcomes by reporting insincere preferences, commonly referred to as strategic voting, has been studied intensely in the social choice literature since the seminal work of Gibbard [Gib73] and Satterthwaite [Sat75] (see also [DS00, EKM07]). The celebrated Gibbard-Satterthwaite Theorem states that essentially all natural voting systems are in some sense manipulable by a strategic voter.

This is bad news. But there is hope! Imagine a voting system that, though manipulable in

¹Notwithstanding the success of approval voting in many such societies, Brams and Fishburn [BF05] also note that approval voting “is not currently used in any public elections, despite efforts to institute it, so its success should be judged as mixed.” For example, the U.S. presidential election, which involves the U.S. electoral college and has aspects of both plurality and majority voting, is not held under approval voting.

principle, has the property that it would confront a strategic voter with a really hard problem to solve when attempting to find a successful manipulative preference to reveal—or even to tell if in the given election such a manipulation is possible. Such a voting system would still be susceptible to manipulation in theory, but one could reasonably hope that due to the complexity of the associated problem no manipulation attempts would ever occur in real elections (or, if they did occur, that they wouldn’t be successful).

Employing computational complexity as a means to protect voting systems from manipulation and other attacks known as “control” attacks was first proposed by Bartholdi, Orlin, Tovey, and Trick [BTT89a,BO91,BTT92] and has since been studied for a wide range of concrete systems (see, e.g., [CS03,EL05,CS06,FHH06a,FHH06b,CSL07,HH07,PR07,MPRZ08,FHS08,ZPR08,BFH⁺08] for results on manipulation, and see Section 3 for the literature regarding control attacks). Some systems have been shown to be “resistant” to manipulation (i.e., informally speaking, their manipulation problem is computationally hard) and some systems have turned out to be “vulnerable” to manipulation (i.e., their manipulation problem is easy to solve).

What does it mean for a problem to be “computationally hard” or “easy” to solve? Complexity theory provides the notions and techniques needed to answer these questions. Two important tasks in complexity theory are to classify problems in terms of their algorithmic complexity and to compare the complexities of two given problems via reductions. This paper will present numerous concrete reductions that establish the computational hardness of problems related to approval voting. For more background on complexity theory, we refer to the textbooks [Pap94,HO02,Rot05].

For approval voting and manipulation, however, the bad news persists: Approval voting is vulnerable to manipulation, even if we allow voters to have “weights” (for example, one approval of a voter of weight five counts as much as five weight-one approvals) and even if we allow a coalition of strategic voters to coordinate their manipulative efforts (as opposed to having a single strategic voter). That is, even in such more general settings, there is a polynomial-time algorithm that solves the manipulation problem for AV. Let us state this problem in the form that will be used for representing computational problems throughout this paper:

Name: AV-MANIPULATION.

Given: A set C of candidates, a list V of ballots over C (that already have been cast by the non-strategic voters, where each ballot gives that voter’s approvals/disapprovals for each $c \in C$) along with these voters’ weights, a list S with the weights of the strategic voters (whose ballots haven’t been cast yet), and a distinguished candidate $p \in C$.

Question: Is it possible for the strategic weighted voters to cast their ballots in such a way that p

becomes an approval winner of the resulting election?

As noted by Faliszewski, Hemaspaandra, and Hemaspaandra [FHH06a,FHH06b],² one can use the following simple greedy strategy to solve this problem in polynomial time: The strategic voters simply approve of their favorite candidate p and disapprove of all other candidates. If this manipulation makes p an approval winner, they have reached their goal (and the polynomial-time algorithm can check whether this happens because, as mentioned earlier, winner determination is easy for AV, and so the algorithm accepts its input in this case). But if p still is not an approval winner after this manipulation, then no strategy whatsoever can turn p into an approval winner (and so, in this case, the algorithm can safely reject its input).

So, as there is no hope for approval voting to computationally resist manipulation, it's time for a change! Let's change what is being changed in the tampering attempts and how it is being changed—that is, let's change the tampering scenario from manipulation to either bribery or control. And let's also change who performs these changes in the election: Both these scenarios differ from manipulation in that they model situations where external actors do seek to affect the outcome.

The model for the complexity-theoretic study of bribery was introduced by Faliszewski, Hemaspaandra, and Hemaspaandra [FHH06a,FHH06b]. In bribery settings, the “briber” seeks to influence the outcome of an election via bribing certain voters to make them change their preferences, without exceeding the briber's budget. The specific bribery scenarios we will consider involve, for example, weighted and unweighted voters, voters with and without price tags, changing one complete ballot (dubbed “bribery”) as opposed to changing just one approval/disapproval in a ballot at unit cost (dubbed “microbribery”), and we will present the associated computational problems and their complexities.

The model for the complexity-theoretic study of control was introduced by Bartholdi, Tovey, and Trick [BTT92]. In control settings, the election's “chair” seeks to influence the outcome of an election via modifying its structure, namely, via actions such as adding/deleting/partitioning either candidates or voters. These scenarios loosely model activities that we often encounter in political elections, such as get-out-the-vote drives (“adding voters”), disenfranchisement and other means of voter suppression (“deleting voters”), introducing new “spoiler” candidates into an election (“adding candidates”), and so on. This paper will present the complexity of AV and SP-AV in each of the common 22 control scenarios, which results in a total of 44 complexity results in this section.

Unfortunately, there is again bad news for AV: Approval voting is vulnerable to many (and is computationally resistant to only four) of these 22 control attacks. However, it will also turn out

²In a different, slightly more flexible scenario, Meir et al. [MPRZ08] showed that approval voting is vulnerable to manipulation by a single strategic voter.

that SP-AV, which stands for “sincere-strategy preference-based approval voting,” displays broad resistance to control: SP-AV is resistant to 19 of the 22 control attacks. That is, among natural voting systems with a polynomial-time winner-determination procedure, SP-AV possesses the broadest resistance to control currently known to hold (see Section 3 for a more detailed discussion that compares SP-AV with other systems displaying an exceptionally broad resistance to control—voting systems that belong to the Copeland/Llull complex of systems [FHHR08b, FHHR08a, FHHR07]).

Unlike many natural and widely used voting systems that are defined in terms of rankings (i.e., strict linear orderings) of the candidates, approval voting merely distinguishes between each voter’s acceptable and unacceptable candidates, yet completely ignores the preference rankings that voters may have about the candidates. To overcome this shortcoming, Brams and Sanver [BS06] proposed SP-AV as a voting system that combines preference-based voting with approval voting. In their definition, they require each voter to have an approval strategy³ that is both “sincere” and “admissible.” An *approval strategy* of a voter is simply a partition of the candidates into approved and disapproved candidates. It is *sincere* if there are no “gaps” (with respect to this voter’s preference ranking of the candidates), i.e., if this voter approves of some candidate then he or she also approves of each candidate ranked higher. A voter’s approval strategy is said to be *admissible* if the voter approves of his or her top candidate and disapproves of his or her bottom candidate. Note that in a 1-candidate election no voter can have an admissible approval strategy.

Sincerity and admissibility are quite natural notions to require. For example, sincerity makes sure that there is no conflict between a voter’s preference ranking and approval strategy. Admissibility in particular prevents approval strategies from being trivial: It is not admissible for a voter to either approve or disapprove of every candidate in an election. Brams and Sanver [BS06] point out that an admissible approval strategy is not dominated in a game-theoretic sense (see Brams and Fishburn [BF78] and also, e.g., Dutta, De Sinopoli, and Laslier [DSL06]). Obviously, if voters are not allowed to either approve or disapprove of everybody then sincere approval strategies are always admissible. Following [ENR08a, ENR08b], we require that all voters must cast only sincere votes and, when there are at least two candidates, voters must cast only sincere, admissible votes (we will call this *Convention 1*).

Within AV, ballots are given as 0-1 (or, equivalently no-yes) approval vectors, where “1” represents approval and “0” disapproval. Within SP-AV, ballots contain more information than this, as they also provide a preference ranking of the candidates. We represent such rankings from left to right (so the leftmost candidate is the most preferred one), and we additionally indicate the ap-

³To avoid confusion, we stress that the “strategy” in “approval strategy” or “AV strategy” is not meant in the sense of strategic voting, but rather simply refers to which candidates are approved of and which are disapproved of by a voter.

provals/disapprovals by inserting an “approval line” into the ranking, where all candidates to the left of this line are approved and all candidates to the right of this line are disapproved. For example, “ $a \ b \ c \mid d$ ” means that this voter approves of a , b , and c , but disapproves of d . Since we require approval strategies (with at least two candidates) to be admissible, the approval line will never be to the left or to the right of all candidates.

There is a central point regarding admissibility that we need to discuss with respect to our control scenarios. In cases of control by deleting candidates or by partition of candidates or voters (the formal definitions of which will be presented in Section 3), it may happen that admissibility will be lost due to the control action. That is, an originally admissible approval strategy might be changed into an inadmissible one by, for example, deletion of candidates. To prevent this from happening (and to obey our convention), we handle such cases by applying the following rule (that we will call *Rule 1*): “If during or after a control action an election with at least two candidates is obtained such that some voter approves of either all candidates or no candidates, then the approval line in each such voter’s ballot is moved so as to respectively disapprove of that voter’s bottom candidate or approve of that voter’s top candidate.” This rule was introduced by Erdélyi, Nowak, and Rothe ([ENR08b], see also [ENR08a]) in order to preserve (or re-enforce) admissibility under such control actions. So, for example, if candidate d has been deleted from an election containing the ballot $a \ b \ c \mid d$ then, according to this rule, this ballot is being changed into $a \ b \mid c$.

Now, coming back to our original question of why one should bother to study the computational aspects of approval voting, it is important to note that this study is motivated not just from the social choice and political science point of view. Indeed, decision-making and preference aggregation are central tasks in many areas of computer science, and voting procedures are far from being confined to political elections in human societies. They have many applications in computer science, ranging from multiagent systems within artificial intelligence (see, e.g., Ephrati and Rosenschein [ER97]) to the development of recommender systems (see Ghosh et al. [GMHS99]) to aggregating the web-page rankings from various search engines (see Dwork et al. [DKNS01]), to name just a few. These are topics at the heart of the emerging area of *computational social choice*, which is at the interface of social choice theory and computer science and is developing rapidly (see Endriss and Lang [EL06] and Endriss and Goldberg [EG08]).

Of course, the computational aspects of voting systems and their associated problems comprise more than just proving these problems hard or easy to solve. Certainly, if some problem has been shown to be computationally hard, the actors involved (be it strategic voters, the chair, or the briber) will seek to find ways to circumvent this obstacle. Fortunately, computer scientists have developed many approaches for how to cope with the computational hardness of problems in practical appli-

cations, and some of these apply here. In particular, we will present an approximation algorithm and some local search heuristics for “minimax approval voting,” an interesting variant of approval voting that was proposed by Brams, Kilgour, and Sanver [BKS04] (see also [BKS07a,BKS07b]) for the purpose of electing a committee of fixed size.

This paper is organized as follows. Section 2 is a detailed discussion of worries about the model, and presents what we feel is the ideal framing. In Section 3, we will present the 44 control results for AV and SP-AV mentioned above. We will also describe the needed complexity-theoretic notions in a way accessible to readers not familiar with complexity theory. In particular, for each of the reductions constructed, we will give comprehensible examples, and also the problems from which we reduce will be illustrated via examples. Section 4 presents the complexity results for bribery. Finally, in Section 5, we will present local search heuristics for minimax approval voting. (Since this is a survey paper, by “present” we do not mean that the results are due to this paper, and in each case we will point the reader to the source papers in the technical literature.)

2 Discussion of Models for Control of Domain-Sensitive Rules

We now come to a particularly important discussion—of the model and approach to SP-AV under control. The previous section describes the “rule,” Rule 1, that [ENR08b] (see also [ENR08a]) creates to handle the fact that unlike any other election system whose control properties have been studied, SP-AV has a domain restriction that has the property that some (in fact, six) of the 22 common control types can turn legal (i.e., in the domain of SP-AV) inputs into inputs that are not in the domain of SP-AV. On its surface, thus, control analysis simply conflicts with SP-AV elections.

[ENR08b] (see also [ENR08a]) approaches this with a rule, Rule 1, that within the control framework readjusts preferences, plus keep in mind also [ENR08b]’s Convention 1, which itself blocks certain votes from even being legal to cast at all. However, this approach is arguably unsatisfying and may lead readers to think, incorrectly, that they are seeing control results about SP-AV. Treating a preference rewrite rule as if it is part of control is unconvincing since control itself is not about doing anything other than what its definition embraces, and in various settings various rewrite rules could be proposed, all in ways whose justification is not about control but hinges on one’s own subjective notion of what the “natural” correction is to move one back onto the domain of the election systems. After all, the [ENR08a,ENR08b] idea that (no, no, no, no) and (yes, yes, yes, yes), each of which show complete equality among the candidates, should be rewritten to, respectively, (yes, no, no, no) and (yes, yes, yes, no), is a matter of taste and this rewrite is not a part of the notion of SP-AV as Brams and Sanver defined it—(no, no, yes, yes) is for example another

possible rewrite.

In fact, the Brams and Sanver notion of admissibility was not designed with control in mind and in the context of control has restrictive, grotesque effects. In control by adding candidates, if one dislikes the two initial candidates but loves all the spoiler candidates, this approach would force one to cast a vote that approved one of the two disliked candidates, and one would be bound under control to have that approval of a disliked candidate remain—along with new approvals of liked candidates—after spoiler candidates were added.⁴ Again, we stress that Brams and Sanver were not anticipating control to be spliced on top of their system, so in mentioning this we are not criticizing their system. However, we will soon give a system, SP-AV-CTA, that we suggest is the natural way to merge control with the general flavor of SP-AV, and we mention now that our approach will remove the effect just noted above; one will be able to cast a vote that says “I will approve my favorite candidate from among those that end up as active candidates in this control by adding candidates election.”

Of course, returning to the [ENR08a,ENR08b] preference rewrite rule, if the preference rewrite rule is of minor importance to the control results one might think there is no issue here. But in fact, the broadened resistance that [ENR08a,ENR08b] prove for “SP-AV” is due not at all to SP-AV itself, but rather is completely due to the preference rewrite rule [ENR08a,ENR08b] introduce. For this entire paper, for consistency with their work, we will echo their view that modifies control with the preference rewrite rule/convention and speaks of the results as if they were about SP-AV.

However, for posterity, we point out what we feel is a more natural approach. In particular, if our process is going to rewrite preferences by some ad hoc rule, the most intellectually frank way to do that is to openly admit that one’s election system is not SP-AV. Rather, consider the following election system, SP-AV-CTA (*Coerce To Admissible*). The system’s domain will be the same as that of SP-AV except it allows inadmissible votes. And its action on any in-domain input is to treat the admissible votes as themselves, to coerce each inadmissible vote into an admissible vote using precisely the preference rewrite rule of [ENR08a,ENR08b], and to then act precisely as SP-AV

⁴This assumes that regarding Convention 1 one’s notion of the “initial election” in control by adding candidates is the basic candidate set C with no spoiler candidates yet added. This is the natural approach and is consistent with how most papers define this control type: Voters indeed have preferences over $C \cup S$, with S the potential spoiler set, but the base election is with C and then one adds in some candidates (call this view of addition of candidates the *Base-Is- C* view). Read absolutely literally, [ENR08a,ENR08b], by diverging from the standard definition, seem to hint at the approach that the initial election in terms of Convention 1 is over $C \cup S$, and after enforcing Convention 1 on that, one deletes some of S . Call this view of addition of candidates the *Base-Is- $(C \cup S)$* view. We mention that this unusual view would avoid the issue mentioned above, but at the cost of taking a quite counterintuitive view of what the original (“before control”) election is in the addition of candidates case.

would on that input.

SP-AV-CTA is in effect precisely what [ENR08a,ENR08b] are feeding their post-control votes into. We suggest one bring SP-AV-CTA out of the closet, and simply directly study it and its control properties. (Of course, SP-AV-CTA may lose some of the nice game-theoretic or other properties of SP-AV. But if so, that too should be openly faced.) Note that SP-AV-CTA sidesteps the control by adding candidates “two disliked candidates initially” example mentioned earlier, as in SP-AV-CTA, a vote of (no, no, . . . , no) plus one’s preference order over all initial and spoiler candidates has the effect of approving precisely one’s favorite candidate (if there are at least two candidates left in, of course).

As a final comment, we mention that control of SP-AV-CTA is not precisely what the [ENR08a, ENR08b] results formally speak to. There is a very subtle difference in that in SP-AV-CTA the initial votes can be inadmissible (but after control all then-inadmissible votes are interpreted as admissible votes). In contrast, in the [ENR08a,ENR08b] model, the initial votes must already be admissible (because our system is SP-AV and admissibility is part of its domain requirement),⁵ plus after the control action they in effect feed the post-control election to the SP-AV-CTA system. However, this subtle distinction in this particular case seems unlikely to remove any resistances, and so we believe SP-AV-CTA is almost certain to retain 19 resistances. But, in any case, as a model, directly studying SP-AV-CTA, with no restrictions, seems cleaner and crisper than claiming to study the known system SP-AV while in effect really studying effects related precisely to SP-AV-CTA’s departures from SP-AV. Indeed, the attractiveness of SP-AV-CTA is sufficiently compelling that Erdélyi, Nowak, and Rothe plan to adopt the SP-AV-CTA model in their final version.

⁵Even this doesn’t mean as much as it seems. Since, for example, control by adding candidates always takes as its input preferences over the union of the initial and the spoiler candidate set, and [ENR08a,ENR08b] diverge from most papers in that they explicitly state that the input election *is* $(C \cup S, V)$ seems to actually in that control case allow, under what we earlier called the Base-is- $(C \cup S)$ view, inputs that are inadmissible with respect to the original election’s candidate set. Presumably, their rule would coerce the ballots that didn’t via the addition of candidates end up admissible. Under the base-is- $(C \cup S)$ view, their approach would be extremely close or quite possibly even semantically identical to looking at—with no special rules or conventions—control for SP-AV-CTA. However, if one really wants to enforce Convention 1, it seems to us that the Base-is- C view is far more compelling. However, our suggestion is that Convention 1 and Rule 1 be discarded—along with the convoluted structure they form—and one simply define stand-alone election systems, such as SP-AV-CTA, that can be analyzed under the utterly standard, long-studied notion of control. In particular, preference coercions should be internalized into the election system.

3 The Control Complexity of Approval Voting and Sincere-Strategy Preference-Based Approval Voting

3.1 Introduction, Background, and Discussion

In this section, we will present the control complexity of AV and SP-AV for each of the ten classic types of procedural control, and also for an eleventh type that has recently been suggested as the “right” version of one of those ten types. Remarkably, each of these complexities—and there are 44 problems involved here—is known. Our goal here is not to prove each of these 44 results, but rather to make clear what each of the results means, and to provide representative proofs that give the flavor of how one obtains such results.

Recall from the Introduction that by studying control, we are asking how hard it is to determine whether a favored candidate can be made the unique winner (or, in the destructive cases, whether a despised candidate can be precluded from being a unique winner) via a given type of alteration of the structure of the election by the election’s “chair.” The eleven types of control are abstracting many types of actions that occur in the real world, such as voter suppression, get-out-the-vote efforts, candidate recruitment, etc. Ideally, we would hope that our election systems block all such attacks—that each such attack either never succeeds (so-called immunity) or that it is computationally intractable for the chair to find such a successful alteration (indeed, even to tell if one exists—so-called resistance). The case we want to avoid—so-called vulnerability—is that the chair’s task is easy: The chair in polynomial time can determine whether he/she can make a successful alteration.

One might worry that such ideal hardness cannot possibly be achieved. For example, perhaps there is an Arrow-like “impossibility” theorem showing that no election system satisfying some natural, attractive, broadly-satisfied-by-typical-election-systems axioms is, for each of our eleven control attacks, resistant or immune to the attack. The existence of such an impossibility theorem is, on its face, at least plausible. However, recently it was shown that for control such an impossibility claim is itself impossible to obtain—because it is untrue. In particular, Hemaspaandra, Hemaspaandra, and Rothe ([HHR07b], see also [FHHR08b] regarding how to include also the eleventh control type) proved that there is an election system—indeed, an election system whose winner-determination problem is computationally feasible—that is resistant to all of our control types.

Given that result, one might naturally ask why the results of this section—on AV and SP-AV—are even worth doing. After all, [HHR07b] provides a “perfect” election system. But one must think about that in context. That election system indeed is “perfect” in terms of avoiding our control

manipulations. But that election system is not AV or SP-AV, and in the real world, approval voting is very attractive and often used, and so the properties of approval voting are richly worth studying—thus, this book! In fact, the election system of [HHR07b] not only is not AV or SP-AV but indeed is a quite complicated, unnatural construct. What that paper does is to show how to hybridize election systems together in such a way that if any one of the constituent systems is immune or resistant to a given control type, then the hybridized system is immune or resistant to that type. Simply put, the process unions the advantages of all the constituent systems. But the price of doing this is a bulky, complicated election system that—while sufficient to prove the impossibility of obtaining an impossibility theorem—would not be attractive for everyday use.

For this reason, it is natural to study already existing, attractive election systems to determine whether they are highly resistant to control. Our hope, of course, is that the systems we study will be highly resistant (or immune) to control. What we will see in this section, however, is that approval voting is vulnerable to many types of control, as shown by Hemaspaandra, Hemaspaandra, and Rothe [HHR07a]. This is disappointing, and is one weakness of approval voting. On the other hand, approval voting is quite attractive in many ways, and so we certainly are not suggesting that AV’s vulnerability to control means AV should never be used. Rather, in selecting an election system, one must weigh the strengths and weaknesses of the systems one is considering, and then must choose the one that is most appropriate for the use to which one will be putting the election system.

However, if one is a fan both of approval voting and of resistance to control, one will find good news in this section. SP-AV, which itself combines aspects of approval voting with aspects of plurality voting, has been shown by Erdélyi, Nowak, and Rothe ([ENR08b], see also [ENR08a]) to have precisely those resistances possessed by either of those systems. As such, it has a higher number of control resistances than any previously studied natural system. The natural systems that previously had the most proven control resistances were from the Copeland/Llull election system complex [FHHR08a, FHHR07]. (Those systems remain the natural systems with the most proven resistances among voting systems in which, unlike AV and SP-AV, votes are simply a strict ordering over the candidates.)

The rest of this section is organized as follows. Section 3.2 quickly presents the election model, the needed notions from complexity theory, and the notions used to classify the hardness of control. Section 3.3, for each control type, describes the motivation for that control type and the results that hold for that control type for AV and SP-AV elections. Section 3.4 gives some conclusions for control issues related to AV and SP-AV.

3.2 Notions: Elections and Complexity

Our elections will have finite numbers of candidates and voters, but the numbers can vary from input to input. So an election is a pair (C, V) , where C is the candidate set and V is a list of ballots (votes) expressing preferences over C . For AV, each ballot will be a length $\|C\|$ bit-vector, expressing approval or disapproval for each candidate. For SP-AV, as discussed in the Introduction, a ballot will be such a vector along with a strict ordering over C . We will use the terms ballot, vote, and voter relatively interchangeably. Within both AV and SP-AV, for each election (C, V) and for each candidate $c \in C$, we use $score_{(C,V)}(c)$ to denote the number of approvals c receives from voters in V . In the case of weighted voters, each approval of a voter of weight k counts as k weight-one approvals.

In this section, we'll be discussing the complexity of control problems. The control types will be presented in Section 3.3, but for now it will suffice to say that for each control-type problem we study, the input to the problem will be an election, (C, V) , a specified candidate, p , and in some cases some information specific to the control problem.

For each control type, we will study two different problems, the constructive problem (trying to make p the unique winner) and the destructive problem (trying to ensure that p ends up not being a unique winner).

To make this a bit more concrete, we now define explicitly two of our problems: Constructive Control by Adding (a Limited Number of) Candidates (CCAC, for short) and Destructive Control by Adding (a Limited Number of) Candidates (DCAC, for short). For these and all other control types, we will in order to ensure cross-comparability with earlier work take the control-type definitions essentially word-for-word from [FHHR08b], which itself for historical consistency is in general following even earlier papers (except [FHHR08b] studies both the model in which we seek to make/preclude a winner and in which we seek to make/preclude a unique winner, but in this paper, like the papers we are most focused on ([HHR07a, ENR08b, ENR08a]), we focus on the unique-winner model). “ $\mathcal{E}$ ” denotes the election system's name (AV or SP-AV in our case).

Control by Adding a Limited Number of Candidates

Name: $\mathcal{E}$ -CCAC and $\mathcal{E}$ -DCAC.

Given: Disjoint sets C and D of candidates, a collection V of voters represented via their ballots over the candidates in the set $C \cup D$, a distinguished candidate $p \in C$, and a nonnegative integer k .

Question ($\mathcal{E}$ -CCAC): Is there a subset E of D such that $\|E\| \leq k$ and p is the unique winner of the $\mathcal{E}$ election $(C \cup E, V)$?

Question ($\mathcal{E}$ -DCAC): Is there a subset E of D such that $\|E\| \leq k$ and p is not a unique winner of the $\mathcal{E}$ election $(C \cup E, V)$?

Now, we are ready to define the notions that are used to describe the difficulty (and possibility) of a given type of control for a given election system. If for a given control type and election system the “chair”—the actor exerting the given type of control—can never, on any input, change p from not being a unique winner to being the unique winner (or, for the destructive case, change p from being the unique winner to not being a unique winner), we say the system is *immune* to the given type of control. One might think that immunity never occurs, but in fact it does. For example, in AV, if p already is losing to (or tying for winner with) some other candidate q , then adding additional candidates certainly won’t make p become the unique winner. Thus we have our first theorem: AV is immune to constructive control by adding candidates [HHR07a]. If immunity does not hold for a given control type, we say the election system is *susceptible* to this type of control.

Regarding election systems susceptible to a type of control, there are two cases that will particularly interest us. If the given problem has a polynomial-time algorithm to tell whether on a given input there exists a control action of the problem’s sort that achieves the chair’s goal for p , we say the election system is *vulnerable* to the given type of control. (In each case in this paper where vulnerability is asserted, something even stronger in fact holds, as noted by the original papers, namely, there is a polynomial-time algorithm that tells whether a successful control action by the chair exists, and if so *produces such an action*.) However, if a given control problem is NP-hard (a central notion in complexity theory to be defined in the next paragraph), we say that the election system is *resistant* to the given type of control. Informally put, resistance to some control type means that the corresponding control problem is impracticable (although potentially not impossible) to solve.

Some readers of this paper may not be familiar with the notion of being “NP-hard,” so we briefly discuss the concept. NP, nondeterministic polynomial time, is the class of all problems that can be solved in polynomial time on a nondeterministic Turing machine. However, an elegant way to describe the class without having to introduce nondeterminism or Turing machines is that NP is the class of all languages L for which there exist a polynomial q and a polynomial-time computable boolean predicate R such that $L = \{x \mid (\exists y)[|y| \leq q(|x|) \text{ and } R(x, y)]\}$. NP is a tremendously important class—it captures the complexity of thousands of crucial problems, ranging from satisfiability of boolean formulas to the traveling salesperson problem. Of course, some NP problems are computationally simple, e.g., the empty set is in NP. The problems that have the property

that each NP problem can be rephrased in terms of them are called the NP-*hard* problems. Formally, a problem A is NP-hard if $(\forall B \in \text{NP})[B \leq_m^P A]$. $B \leq_m^P A$ (“ B many-one polynomial-time reduces to A ”) by definition means there exists a polynomial-time computable function f such that $(\forall x)[x \in B \iff f(x) \in A]$. A problem is called NP-*complete* exactly if it is in NP and is NP-hard. Each of the problems stated in this paper as being resistant in fact is not only NP-hard, but also happens to be NP-complete.

The concepts of immunity, susceptibility, vulnerability, and resistance for control were introduced by Bartholdi, Tovey, and Trick [BTT92], and the above definitions are theirs except we follow the more logical, now more common approach, introduced in [HHR07b], of defining resistant as being “susceptible, NP-hard” problems ([BTT92] defined it as “susceptible, NP-complete” problems).

3.3 44 Control Results

In this section, for each of the eleven commonly discussed types of control, and for each of the constructive and destructive cases, we present what results hold for AV and SP-AV—44 results in all. We’ll do so by going right through the types (sometimes in groups), mentioning the intuition/motivation for the type (when doing so we will—unless we explicitly mention otherwise—be giving the intuition for the constructive case, but from that one can naturally see the intuition for the destructive case), and then will state what results hold. For a handful of cases—enough to give the reader the flavor of how immunity, susceptibility, vulnerability, and resistance are proven—we will include proofs. (Readers not interested in how such results are proven will want to skip over such proofs.)

To collect and have the results all in one place, Table 1 presents all 44 results.

3.3.1 Adding and Deleting Candidates

We already defined control by adding a limited number of candidates. We now define the remaining two types of control by candidate addition/deletion.

Control by Adding an Unlimited Number of Candidates

Name: $\mathcal{E}$ -CCAC_u and $\mathcal{E}$ -DCAC_u.

Given: Disjoint sets C and D of candidates, a collection V of voters represented via their ballots over the candidates in the set $C \cup D$, and a distinguished candidate $p \in C$.

Control by	SP-AV		AV	
	Constructive	Destructive	Constructive	Destructive
Adding an Unlimited Number of Candidates	R	R	I	V
Adding a Limited Number of Candidates	R	R	I	V
Deleting Candidates	R	R	V	I
Partition of Candidates	TE: R TP: R	TE: R TP: R	TE: V TP: I	TE: I TP: I
Run-off Partition of Candidates	TE: R TP: R	TE: R TP: R	TE: V TP: I	TE: I TP: I
Adding Voters	R	V	R	V
Deleting Voters	R	V	R	V
Partition of Voters	TE: R TP: R	TE: V TP: R	TE: R TP: R	TE: V TP: V

Table 1: Overview of results. Key: I means immune, R means resistant, V means vulnerable, TE means ties-eliminate, and TP means ties-promote. Results for SP-AV are due to Erdélyi, Nowak, and Rothe [ENR08b]; their proofs are either due to [ENR08b] or draw on proofs from [HHR07a]. Results for AV are due to Hemaspaandra, Hemaspaandra, and Rothe [HHR07a]. (The results for control by adding a limited number of candidates for AV, though not stated explicitly in [HHR07a], follow immediately from the proofs of the corresponding results for the “unlimited” variant of the problem.)

Question ($\mathcal{E}$ -CCAC_u): Is there a subset E of D such that p is the unique winner of the $\mathcal{E}$ election $(C \cup E, V)$?

Question ($\mathcal{E}$ -DCAC_u): Is there a subset E of D such that p is not a unique winner of the $\mathcal{E}$ election $(C \cup E, V)$?

Control by Deleting Candidates

Name: $\mathcal{E}$ -CCDC and $\mathcal{E}$ -DCDC.

Given: A set C of candidates, a collection V of voters represented via their ballots over C , a distinguished candidate $p \in C$, and a nonnegative integer k .

Question ($\mathcal{E}$ -CCDC): Is it possible to by deleting at most k candidates ensure that p is the unique winner of the resulting $\mathcal{E}$ election?

Question ($\mathcal{E}$ -DCDC): Is it possible to by deleting at most k candidates other than p ensure that p is not a unique winner of the resulting $\mathcal{E}$ election?

Following [FHR08b], we consider “limited” candidate addition the more natural addition notion, but we mention that “unlimited” candidate addition is what (along with the same other nine notions we will present here—except for the handling of ties in the subelections of the control-by-partition cases, see Section 3.3.2) was used by Bartholdi, Tovey, and Trick [BTT92].

The three add/delete candidate control types model candidate recruitment and forcing candidates out of the race. For example, Ralph Nader joining the U.S. presidential contest may have helped George W. Bush in his race against Al Gore. Similarly, if once in the race Ralph Nader could have been persuaded to drop out, that might have been helpful to Al Gore.

The results that hold for these control types are as follows. AV is immune to constructive control by both limited and unlimited addition of candidates, and the argument we gave about this earlier in this paper suffices to prove both those cases. Essentially the same reasoning shows that AV is immune to destructive control by deletion of candidates—if p is already the unique winner under AV, and so is approved by more voters than is any other candidate, then deleting candidates (other than p) will clearly leave p still the unique winner, since p still will have more approvals. So, we have shown the following.

Theorem 3.1 ([HHR07a]) *AV is immune to constructive control both by adding a limited and by adding an unlimited number of candidates, and to destructive control by deleting candidates.*

However, if one considers the same control cases as in Theorem 3.1 but with “constructive” and “destructive” being swapped, one obtains vulnerability rather than immunity results.

Theorem 3.2 ([HHR07a]) *AV is vulnerable to destructive control both by adding a limited and by adding an unlimited number of candidates, and to constructive control by deleting candidates.*

In contrast, SP-AV is resistant to control in these six cases. As an example of proving susceptibility and indeed resistance, we give a proof of the result of [ENR08b] that SP-AV is resistant to constructive control by deleting candidates.

Theorem 3.3 ([ENR08b]) *SP-AV is resistant to constructive control by deleting candidates.*

Proof. The reason why SP-AV is susceptible to constructive control by deleting candidates is quite simple: SP-AV is a “voiced” voting system (i.e., SP-AV has the property that in each single-candidate election, this candidate—even with zero approvals—wins), and it is known that every voiced voting system is susceptible to constructive control by deleting candidates [HHR07a]. Indeed, consider an election with exactly two candidates, p and q , in which p and q tie for winner, i.e.,

they have the same number of approvals. So p is not a unique winner in this election, but deleting q from it makes p the unique SP-AV winner of the resulting election.

To prove that SP-AV is resistant to constructive control by deleting candidates, we need to show that the corresponding control problem, SP-AV-CCDC, is NP-hard (which, as mentioned in Section 3.2, means that each NP problem is $\leq_m^P$ -reducible to SP-AV-CCDC). Fortunately, we do not have to give a $\leq_m^P$ -reduction from every single NP problem to this problem; rather, it is enough to reduce just one NP-complete problem to SP-AV-CCDC.⁶

We choose HITTING SET as our NP-complete problem (see, e.g., Garey and Johnson [GJ79]) from which to reduce. Before we formally define this problem, we give an illustrative explanation from everyday life. Consider a set, B , of m students. Each student has chosen to sign up for some subset of n courses, where we assume that courses for which no students signed up have already been cancelled. So, the i th course, $1 \leq i \leq n$, is attended by some nonempty subset of the students, say $S_i \subseteq B$. Of course, students may have signed up for different courses and some lazy students may have chosen to sign up for no course at all. The professors who teach these courses are exceedingly busy doing research, which is why they want to pay a student in each of the n courses to help them produce a course report for that course (to be given to the Dean). However, since the departmental budget is limited, they can afford to pay only $k < m$ students. Also, the limit on the departmental budget is the reason why students will get one and the same amount of money for their help, no matter how many course reports they help to write. We further assume that the students in B (except the lazy ones, of course, who haven't signed up for any course) are so eager to help that they would volunteer to do the job even if they had to write all n course reports. The task the professors face is to determine whether there is some set of at most k students in B such that each course is attended by at least one of them—in other words, they need to determine whether there exists a hitting set of size at most k for $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$. A bit more formally phrased, this problem is defined as follows:

Name: HITTING SET.

Given: A set $B = \{b_1, b_2, \dots, b_m\}$, a collection $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$ of nonempty subsets S_i of B , and a positive integer $k < m$.

Question: Does there exist a hitting set of size at most k for $\mathcal{S}$, i.e., does there exist a subset $B' \subseteq B$ such that $\|B'\| \leq k$ and for each i , $1 \leq i \leq n$, we have $S_i \cap B' \neq \emptyset$?

⁶This follows by transitivity of the $\leq_m^P$ -reducibility: Each NP problem can be $\leq_m^P$ -reduced to any NP-complete problem, so if we find a $\leq_m^P$ -reduction from some NP-complete problem to SP-AV-CCDC, then each NP problem $\leq_m^P$ -reduces to SP-AV-CCDC.

Let $(B, \mathcal{S}, k)$ be a given instance of HITTING SET, where $B = \{b_1, b_2, \dots, b_m\}$, $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$ with $\emptyset \neq S_i \subseteq B$ for each i , $1 \leq i \leq n$, and $k < m$ is a positive integer.⁷ To prove $\text{HITTING SET} \leq_m^p \text{SP-AV-CCDC}$, we now define a reduction that in polynomial time transforms $(B, \mathcal{S}, k)$ into an instance $(C, V, p, m - k)$ of SP-AV-CCDC such that the HITTING SET question for $(B, \mathcal{S}, k)$ is answered “yes” if and only if the SP-AV-CCDC question for $(C, V, p, m - k)$ is answered “yes.”

Our instance $(C, V, p, m - k)$ of SP-AV-CCDC is constructed from the given HITTING SET instance $(B, \mathcal{S}, k)$ as follows. The candidate set is $C = B \cup \{p\}$. There are $4n(k + 1) + 4m - 2k + 3$ voters in V whose ballots belong to the following five groups:

1. The first group contains, for each i with $1 \leq i \leq n$, $2(k + 1)$ ballots of the following type (this notation will be explained more clearly below):

$$S_i \mid (B - S_i) \mid p.$$

2. The second group contains, for each i with $1 \leq i \leq n$, $2(k + 1)$ ballots of the following type:

$$(B - S_i) \mid p \mid S_i.$$

3. The third group contains, for each j with $1 \leq j \leq m$, two ballots of the following type:

$$b_j \mid p \mid (B - \{b_j\}).$$

4. The fourth group contains $2(m - k)$ ballots of the following type:

$$B \mid p.$$

5. The fifth group contains three ballots of the following type:

$$p \mid B.$$

In the above notation, each time a set is written in a ballot it is a shorthand for having its elements listed in the same order that they occur in the list $(b_1, b_2, \dots, b_m)$. For example, if $B = \{b_1, b_2, \dots, b_6\}$ contains six members and $S_3 = \{b_2, b_3, b_5\}$ contains three, then “ $S_3 \mid (B - S_3) \mid p$ ” means that each of the $2(k + 1)$ ballots corresponding to S_3 has the form: $b_2 \ b_3 \ b_5 \mid b_1 \ b_4 \ b_6 \ p$.

⁷HITTING SET is sometimes defined with “ $k \leq m$ ” instead of “ $k < m$.” However, since B is always a hitting set of size k if $m = k$ (due to $\mathcal{S}$ containing no empty sets), we may require $k < m$ in the problem instance.

$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	$b_1 \mid p \ b_2 \ b_3 \ b_4,$
$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	$b_1 \mid p \ b_2 \ b_3 \ b_4,$
$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	
$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	$b_2 \mid p \ b_1 \ b_3 \ b_4,$
$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	$b_2 \mid p \ b_1 \ b_3 \ b_4,$
$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	
		$b_3 \mid p \ b_1 \ b_2 \ b_4,$
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	$b_3 \mid p \ b_1 \ b_2 \ b_4,$
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	$b_4 \mid p \ b_1 \ b_2 \ b_3,$
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	$b_4 \mid p \ b_1 \ b_2 \ b_3,$
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
		$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	$p \mid b_1 \ b_2 \ b_3 \ b_4,$
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	$p \mid b_1 \ b_2 \ b_3 \ b_4,$
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	$p \mid b_1 \ b_2 \ b_3 \ b_4.$

Table 2: 51 ballots generated from a “yes” instance of HITTING SET.

It is obvious that $(C, V, p, m - k)$ can be computed from $(B, \mathcal{S}, k)$ in polynomial time.

Note that, regardless of whether $(B, \mathcal{S}, k)$ is a “yes” or a “no” instance of HITTING SET, p is not the unique SP-AV winner (and not even *an* SP-AV winner) of the election (C, V) constructed. In particular, note that p is approved by $2n(k + 1) + 3$ voters in (C, V) , but every $b_i \in B$ has $2n(k + 1) + 2 + 2(m - k)$ approvals, and since $k < m$, every $b_i \in B$ is better off than p .

On the other hand, we will show that if $\mathcal{S}$ has a size- k hitting set then p can be made the unique SP-AV winner by deleting at most $m - k$ candidates, and if $\mathcal{S}$ does not have a hitting set of size k then this is not possible. But before we proceed with the proof, let us first look at an illustrative example of the construction.

Example 3.1 Suppose we are given the HITTING SET instance $(B, \mathcal{S}, 2)$, where $B = \{b_1, b_2, b_3, b_4\}$

and $\mathcal{S} = \{S_1, S_2, S_3\}$ with $S_1 = \{b_1, b_2\}$, $S_2 = \{b_1, b_4\}$, and $S_3 = \{b_3, b_4\}$. Our reduction then yields the instance $(C, V, p, 2)$ of SP-AV-CCDC with candidate set $C = \{b_1, b_2, b_3, b_4, p\}$ and with V consisting of the 51 ballots shown in Table 2.

The ballots in the left column correspond to the first group, those in the middle column to the second group, and the ballots in the right column correspond to the third, fourth, and fifth groups. Note that $(B, \mathcal{S}, 2)$ is a “yes” instance of HITTING SET: Choose, for example, the two-element set $B' = \{b_1, b_4\}$ and note that B' has a nonempty intersection with each of S_1 , S_2 , and S_3 , so B' is a hitting set of size two for $\mathcal{S}$.

Our construction ensures that $(C, V, p, 2)$ is also a “yes” instance of SP-AV-CCDC. In (C, V) , p is approved by 21 voters and each $b_i \in B$ is approved by 24 voters, so p does not win. However, p can be made the unique SP-AV winner by deleting two candidates from C . In particular, if we delete b_2 and b_3 from B (i.e., if we delete all candidates not in the hitting set B' mentioned above), we obtain the election with candidate set $B' \cup \{p\} = \{b_1, b_4, p\}$ and voters in V restricted to $B' \cup \{p\}$. By the rule (Rule 1) about moving the approval line to ensure admissibility, p gains four approvals in this election, namely, p gains two approvals by changing the two ballots of the form $b_2 \mid p \ b_1 \ b_3 \ b_4$ into $p \mid b_1 \ b_4$ and two approvals by changing the two ballots of the form $b_3 \mid p \ b_1 \ b_2 \ b_4$ into $p \mid b_1 \ b_4$. Since p doesn’t lose any of the approvals it already had before the deletion, p now has 25 approvals. However, both b_1 and b_4 still have 24 approvals, so p is now the unique SP-AV winner.

Now, consider the same HITTING SET instance as above except with the parameter k decreased by one: $(B, \mathcal{S}, 1)$. Note that $(B, \mathcal{S}, 1)$ is a “no” instance of HITTING SET: No one-element subset of B hits each of S_1 , S_2 , and S_3 , i.e., for no i , $1 \leq i \leq 4$, does $\{b_i\}$ have a nonempty intersection with each of S_1 , S_2 , and S_3 .

Starting from $(B, \mathcal{S}, 1)$, our reduction yields the SP-AV-CCDC instance $(C, V', p, 3)$ with the same candidate set as above, $C = \{b_1, b_2, b_3, b_4, p\}$, and with the new V' consisting of the 41 ballots shown in Table 3.

To see that $(C, V', p, 3)$ is a “no” instance of SP-AV-CCDC, first note that p has 15 approvals in (C, V') , yet each $b_i \in B$ is approved by 20 voters, so p does not win. Moreover, it can be checked that deleting any subset of at most three candidates from B will leave at least one $b_i \in B$ in the race that has more approvals than p .

It is not a coincidence that, as in Example 3.1, “yes” (respectively, “no”) instances of HITTING SET yield “yes” (respectively, “no”) instances of SP-AV-CCDC. Indeed, continuing with the proof of Theorem 3.3, we will now show that, in general, B has a hitting set of size k for $\mathcal{S}$

$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	$b_1 \mid p \ b_2 \ b_3 \ b_4,$
$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	$b_1 \mid p \ b_2 \ b_3 \ b_4,$
$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	$b_2 \mid p \ b_1 \ b_3 \ b_4,$
$b_1 \ b_2 \mid b_3 \ b_4 \ p,$	$b_3 \ b_4 \ p \mid b_1 \ b_2,$	$b_2 \mid p \ b_1 \ b_3 \ b_4,$
		$b_3 \mid p \ b_1 \ b_2 \ b_4,$
		$b_3 \mid p \ b_1 \ b_2 \ b_4,$
		$b_4 \mid p \ b_1 \ b_2 \ b_3,$
		$b_4 \mid p \ b_1 \ b_2 \ b_3,$
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
$b_1 \ b_4 \mid b_2 \ b_3 \ p,$	$b_2 \ b_3 \ p \mid b_1 \ b_4,$	$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
		$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
		$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
		$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
		$b_1 \ b_2 \ b_3 \ b_4 \mid p,$
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	$p \mid b_1 \ b_2 \ b_3 \ b_4,$
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	$p \mid b_1 \ b_2 \ b_3 \ b_4,$
$b_3 \ b_4 \mid b_1 \ b_2 \ p,$	$b_1 \ b_2 \ p \mid b_3 \ b_4,$	$p \mid b_1 \ b_2 \ b_3 \ b_4.$

Table 3: 41 ballots generated from a “no” instance of HITTING SET.

if and only if p can be made the unique SP-AV winner by deleting at most $m - k$ candidates from the election (C, V) .

Suppose there is a hitting set B' of size k for $\mathcal{S}$. Since deleting the $m - k$ candidates in $B - B'$ from C will move the approval line—due to the rule, Rule 1, of re-enforcing admissibility—for $2(m - k)$ voters of the third group, p will gain these approvals of the candidates in $B - B'$. Thus in $(B' \cup \{p\}, V)$, p is now approved by $2n(k + 1) + 2(m - k) + 3$ voters and each member of B' still has $2n(k + 1) + 2 + 2(m - k)$ approvals, so p is the unique SP-AV winner of the election $(B' \cup \{p\}, V)$.

Conversely, suppose that p can be made the unique SP-AV winner by deleting no more than $m - k$ candidates from C . Of course, p cannot have been deleted, so the deleted candidates all belong to B . Let B' be the set of candidates from B remaining in the race after deletion. Since

$\|B - B'\| \leq m - k$, we have $\|B'\| \geq k$. We will show that B' is a hitting set of size exactly k for $\mathcal{S}$. Let h be the number of sets S_i in $\mathcal{S}$ not hit by B' , i.e., $h = \|\{i \mid S_i \in \mathcal{S} \text{ and } B' \cap S_i = \emptyset\}\|$.

In $(B' \cup \{p\}, V)$, p is approved by

$$(1) \quad 2(n - h)(k + 1) + 2(m - \|B'\|) + 3$$

voters. From our assumption that all candidates occur in the same ordering in each of the $2n(k + 1)$ first-group voters (those of the form $S_i \mid (B - S_i) \mid p$, where $1 \leq i \leq n$), it follows that for each S_i not hit by B' , one and the same candidate in B' (namely, the candidate occurring first in the order of B') receives an additional $2(k + 1)$ approvals—due to moving the approval line—thanks to the $2(k + 1)$ voters corresponding to S_i after deletion of the candidates in $B - B'$. Summing up, this candidate will end up having exactly

$$(2) \quad 2(n + h)(k + 1) + 2 + 2(m - k)$$

approvals after deletion of $B - B'$. Since p has been made the unique SP-AV winner by deleting the candidates in $B - B'$ from C , p must have more approvals in $(B' \cup \{p\}, V)$ than any candidate in B' . So by Equations (1) and (2), we have

$$2(n - h)(k + 1) + 2(m - \|B'\|) + 3 > 2(n + h)(k + 1) + 2 + 2(m - k)$$

or, equivalently,

$$1 + 2(k - \|B'\|) > 4h(k + 1).$$

However, since h is a nonnegative integer satisfying

$$1 = \frac{4(k + 1)}{4(k + 1)} > \frac{1 + 2(k - \|B'\|)}{4(k + 1)} > h,$$

we have $h = 0$, which implies that B' is a hitting set for $\mathcal{S}$. Moreover, $h = 0$ also implies that $1 + 2(k - \|B'\|)$ must be positive, so $\|B'\| \leq k$. Summing up, B' is a hitting set of size k for $\mathcal{S}$. $\square$

3.3.2 The Partition Cases

We now come to a particularly interesting and challenging collection of cases: those cases having to do with partitioning into subelections. In these cases, the election becomes a two-round process. There is a first round, containing one or two subelections depending on the case, and then there is a (final) second round.

Since there is a first round, an issue arises here that we did not have in Section 3.3.1. Namely, what should we do if in a first-round election some candidates *tie* as winners? There are two natural approaches, both introduced in [HHR07a], and since people seem to have strong feelings as to one or the other being more natural—but with no broad agreement among people—many papers take the cautious approach of giving results for both approaches to the issue of whom to move forward from tied first-round subelections. The two approaches are *ties promote* (in Table 1, “TP”), in which all subelection winners move forward to the final round, and *ties eliminate* (in Table 1, “TE”), in which one moves forward from a subelection exactly if one is the one and only winner of that subelection. It is worth remarking that Table 1 contains three cases in which these two rules yield different results. Dealing with ties truly is part of a partition-based control model, and not some unimportant detail that never affects one’s study.

Among the two models, we consider the TE model by far the more consistent with the rest of the framework. The reason is that in control problems (and so, in the second-round elections), we are asking whether candidate p can be made the (or be precluded from being a) *unique* winner. Although control problems are most typically studied in the unique-winner model, sometimes they are studied in the model in which one just asks whether the given candidate is (this is the constructive case) or is not (this is the destructive case) a winner. For example, Faliszewski et al. [FHHR07, FHHR08b] study both models. We mention that, analogously to the comments just made, if one’s model is the nonunique-winner model, then our feeling is that the more natural approach to ties in partition-related subelections is the TP model.

We now state the three partition-based control types.

Control by Run-Off Partition and Partition of Candidates

Name: $\mathcal{E}$ -CCRPC and $\mathcal{E}$ -DCRPC (Control by Run-Off Partition of Candidates).

Given: A set C of candidates, a collection V of voters represented via ballots over C , and a distinguished candidate $p \in C$.

Question ($\mathcal{E}$ -CCRPC): Is there a partition of C into C_1 and C_2 such that p is the unique winner of the two-stage election where the winners of subelection (C_1, V) that survive the tie-handling rule compete against the winners of subelection (C_2, V) that survive the tie-handling rule? Each subelection (in both stages) is conducted using election system $\mathcal{E}$.

Question ($\mathcal{E}$ -DCRPC): Is there a partition of C into C_1 and C_2 such that p is not a unique winner of the two-stage election where the winners of subelection (C_1, V) that survive the tie-handling

rule compete against the winners of subelection (C_2, V) that survive the tie-handling rule? Each subelection (in both stages) is conducted using election system $\mathcal{E}$.

The above description defines four computational problems for a given election system $\mathcal{E}$: $\mathcal{E}$ -CCRPC-TE, $\mathcal{E}$ -CCRPC-TP, $\mathcal{E}$ -DCRPC-TE, and $\mathcal{E}$ -DCRPC-TP.

Name: $\mathcal{E}$ -CCPC and $\mathcal{E}$ -DCPC (Control by Partition of Candidates).

Given: A set C of candidates, a collection V of voters represented via ballots over C , and a distinguished candidate $p \in C$.

Question ($\mathcal{E}$ -CCPC): Is there a partition of C into C_1 and C_2 such that p is the unique winner of the two-stage election where the winners of subelection (C_1, V) that survive the tie-handling rule compete against all candidates in C_2 ? Each subelection (in both stages) is conducted using election system $\mathcal{E}$.

Question ($\mathcal{E}$ -DCPC): Is there a partition of C into C_1 and C_2 such that p is not a unique winner of the two-stage election where the winners of subelection (C_1, V) that survive the tie-handling rule compete against all candidates in C_2 ? Each subelection (in both stages) is conducted using election system $\mathcal{E}$.

This description defines four computational problems for a given election system $\mathcal{E}$: $\mathcal{E}$ -CCPC-TE, $\mathcal{E}$ -CCPC-TP, $\mathcal{E}$ -DCPC-TE, and $\mathcal{E}$ -DCPC-TP.

Control by Partition of Voters

Name: $\mathcal{E}$ -CCPV and $\mathcal{E}$ -DCPV.

Given: A set C of candidates, a collection V of voters represented via ballots over C , and a distinguished candidate $p \in C$.

Question ($\mathcal{E}$ -CCPV): Is there a partition of V into V_1 and V_2 such that p is the unique winner of the two-stage election where the winners of subelection (C, V_1) that survive the tie-handling rule compete against the winners of subelection (C, V_2) that survive the tie-handling rule? Each subelection (in both stages) is conducted using election system $\mathcal{E}$.

Question ($\mathcal{E}$ -DCPV): Is there a partition of V into V_1 and V_2 such that p is not a unique winner of the two-stage election where the winners of subelection (C, V_1) that survive the tie-handling rule compete against the winners of subelection (C, V_2) that survive the tie-handling rule? Each subelection (in both stages) is conducted using election system $\mathcal{E}$.

The motivation for run-off partition of candidates is any process, perhaps a legislative process, in which a given committee considers two batches of alternatives, votes on each separately, and then considers only the winners. For example, a physics department’s faculty search might result in many experimental and many theoretical candidates, and the department chair might try to clarify the decision process by having the entire department faculty vote separately among the experimental candidates and among the theoretical candidates, and after that limit the discussion and final-round vote to only those candidates that survived the first-round vote.

In contrast, (non-run-off) partition of candidates models systems in which there is a qualifying election that some candidates are exempted from. For example, in some tournaments, the host country may be given such an exemption from qualification.

Finally, partition of voters models a primary system in which the electorate is divided into (two) groups, each group votes over the candidates, and the winners move forward to a final election in which everyone votes. In our previous example, this would be the case if the department had the theoreticians hold an election over all candidates and separately had the experimentalists hold an election over all candidates—yes, one can make some guesses about who might win in each subelection!—and then had both groups jointly vote over those candidates surviving the first round.

The results that hold for the partition cases are as follows. For AV, resistance holds for constructive control by partition of voters (both TP and TE), vulnerability holds for the TE cases of constructive partition of candidates, constructive run-off partition of candidates, and destructive partition of voters, vulnerability also holds for the TP case of destructive partition of voters, and immunity holds in the remaining six cases. For SP-AV, vulnerability holds for the TE case of destructive control by partition of voters, and resistance holds for the other eleven cases.

To illustrate vulnerability proofs, we prove the following result.

Theorem 3.4 ([HHR07a]) *Approval voting is vulnerable to destructive control by partition of voters in the TE model.*

Proof. First, we give an example showing that approval voting is susceptible to destructive control by partition of voters in the TE model. Consider the election (C, V) with candidate set $C = \{a, b, c\}$, distinguished candidate c , and the following collection $V = \{v_1, v_2, \dots, v_{17}\}$ of ballots, each being represented as an approval vector for abc in $\{0, 1\}^3$: $v_1 = \dots = v_5 = 100$, $v_6 = 110$, $v_7 = \dots = v_{10} = 010$, and $v_{11} = \dots = v_{17} = 001$. Clearly, c is the unique approval winner in (C, V) . However, if we partition V into $V_1 = \{v_1, \dots, v_5, v_{11}, \dots, v_{14}\}$ and $V_2 = V - V_1$, then candidate a is the unique approval winner of the first-round subelection (C, V_1) and candidate b is the unique approval winner of the other first-round subelection, (C, V_2) . Since c doesn’t proceed to the final round, c ’s victory

has been successfully blocked by voter partition. Thus approval voting is susceptible to destructive control by partition of voters in the TE model.

To prove vulnerability, we now describe a polynomial-time algorithm for the control problem at hand. The algorithm takes as input an election (C, V) and a distinguished candidate $p \in C$. The output of the algorithm either will be a successful voter partition (V_1, V_2) (i.e., a partition such that p won't be a unique approval winner in the final round of the corresponding two-stage election), or will be “control impossible” (this will be the case exactly if there is no way for the chair to exert a successful control action of this type).

In a nutshell, the basic idea in the algorithm (after having handled certain trivial cases including the case where p is not a unique approval winner of (C, V) and the cases with $\|C\| \leq 2$ —so we now have that $\|C\| \geq 3$ and that p is the unique approval winner of (C, V)) is to check whether it is possible to find two candidates (other than the distinguished candidate p) who can prevent p from proceeding to the final round of the two-stage election induced by some partition of voters. More precisely, if two such candidates indeed can be found, they will preclude p from the final round by tying or defeating p in each of the two first-round subelections. And if they cannot be found, it is impossible to block p 's final-round victory in any partition of voters.

Before describing the algorithm, let us establish some useful notation. Consider (for elections having at least three candidates) any two distinct candidates $a, b \in C - \{p\}$. Our notation will focus on only the approvals/disapprovals of a , b , and p , and will not care about which of the other candidates are approved of. For $x \in \{a, b\}$, define the following six sets:

$$\begin{aligned} S_x &= \{v \in V \mid v \text{ approves of } x \text{ and disapproves of both } p \text{ and the candidate in } \{a, b\} - \{x\}\}; \\ S_{xp} &= \{v \in V \mid v \text{ approves of both } x \text{ and } p \text{ and disapproves of the candidate in } \{a, b\} - \{x\}\}; \\ W_p &= \{v \in V \mid v \text{ approves of } p \text{ and disapproves of both } a \text{ and } b\}; \\ L_p &= \{v \in V \mid v \text{ disapproves of } p \text{ and approves of both } a \text{ and } b\}. \end{aligned}$$

We now describe our algorithm. The algorithm first checks the following trivial cases:

1. If C contains p alone, then output “control impossible” and stop. (There is no other candidate who could possibly prevent p from winning.)
2. Else if p is not a unique approval winner, then output the successful partition $(V, \emptyset)$ and stop.
3. Else if $\|C\| = 2$, then output “control impossible” and stop. (In this case, p wins at least one of the subelections—in whatever partition of voters the chair chooses—and proceeds to the final stage, where it basks in glory as the unique approval winner.)

Now, if none of the trivial cases apply, we know that p currently is the unique approval winner in (C, V) and C has at least two members other than p . To determine whether p can be dethroned as the unique approval winner, the algorithm enters a loop to check for each pair of distinct candidates $a, b \in C - \{p\}$ whether

$$(3) \quad \|W_p\| > \|S_a\| + \|S_b\| + \|L_p\|$$

is satisfied. If so, p has so many more approvals than this a and b that this a and b are helpless against p in any voter partition, and the algorithm enters the next loop iteration to check the next a and b . As soon as some a and b are found for which $\|W_p\| \leq \|S_a\| + \|S_b\| + \|L_p\|$ is satisfied (i.e., (3) does not hold), it outputs the successful partition (V_1, V_2) and stops, where $V_1 = S_a \cup S_{ap} \cup W'_p$ with W'_p containing the first $\min(\|S_a\|, \|W_p\|)$ voters of W_p , and where V_2 is $V - V_1$. If all possible pairs of candidates (other than p) have been checked and none has produced a successful partition of V , the algorithm outputs “control impossible” and stops.

Why does this algorithm correctly decide whether destructive control by partition of voters in model TE is possible for AV? On the one hand, if (3) is satisfied for all pairs of candidates a and b then, for whatever voter partition is chosen, no pair of candidates can tie or defeat p in *both* subelections, so p will be the unique approval winner of at least one subelection and will thus proceed to the final round, which it alone will win. On the other hand, if there exists a candidate pair a and b for which (3) fails to hold then the algorithm partitions V into (V_1, V_2) as stated above. It follows that, in (C, V_1) , we have

$$\text{score}_{(C, V_1)}(a) - \text{score}_{(C, V_1)}(p) = \|S_a\| - \min(\|S_a\|, \|W_p\|) \geq 0,$$

which means that a ties or defeats p in (C, V_1) . To show that b also ties or defeats p in the other subelection, (C, V_2) , we have to show that

$$\text{score}_{(C, V_2)}(b) - \text{score}_{(C, V_2)}(p) = \|S_b\| + \|L_p\| - (\|W_p\| - \min(\|S_a\|, \|W_p\|)) \geq 0.$$

Transform this inequality into the form

$$\|W_p\| - \|L_p\| \leq \min(\|S_a\|, \|W_p\|) + \|S_b\|.$$

If $\|W_p\| < \|S_a\|$, we have $\|W_p\| - \|L_p\| \leq \|W_p\| + \|S_b\|$, which is always true because both $\|L_p\|$ and $\|S_b\|$ are nonnegative. If $\|W_p\| \geq \|S_a\|$, however, we have $\|W_p\| \leq \|S_a\| + \|S_b\| + \|L_p\|$, which is true because (3) does not hold in the current case. Thus this algorithm correctly decides AV-DCPV for model TE in polynomial time. $\square$

As an example of a resistance proof based on a reduction from the “X3C” problem (which will be defined in a minute), we now prove the following result.

Theorem 3.5 ([HHR07a]) *Approval voting is resistant to constructive control by partition of voters in the TP model.*

Proof. To prove susceptibility, consider the same example as in the proof of Theorem 3.4, except that now a instead of c is the distinguished candidate. Initially, a is not a unique approval winner of (C, V) , but the same partition of V that was described in the proof of Theorem 3.4 does make a the unique approval winner. Thus, approval voting is susceptible to constructive control by partition of voters in model TP.

To prove resistance, we reduce from the following problem.

Name: EXACT COVER BY THREE-SETS (X3C, for short).

Given: A set $B = \{b_1, b_2, \dots, b_{3m}\}$, $m \geq 1$, and a collection $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$ of sets such that for each i , $1 \leq i \leq n$, it holds that $S_i \subseteq B$ and $\|S_i\| = 3$.

Question: Does there exist an exact cover for B , i.e., does there exist a set $A \subseteq \{1, 2, \dots, n\}$, $\|A\| = m$, such that $\bigcup_{i \in A} S_i = B$?

To give an illustrative (even though, admittedly, not really everyday) example of this problem, we mention the infamous theft of the splendid glass mosaic covering the interior of the Cathedral of Monreale on Sicily, which was created by artists from Sicily and Constantinople in the twelfth century. One morning, the bishop entered the cathedral and, to his utter dismay, every single one of the 15,000,000 mosaic pieces, which had covered about 6,340 square meters, was gone! Let us enumerate the stolen mosaic pieces and call the mosaic $B = \{b_1, b_2, \dots, b_{15,000,000}\}$, so $m = 5,000,000$ in this example. The police started an intense search for the evildoers and the lost treasure. Soon thereafter, a number of pieces of this invaluable mosaic—surprisingly always in batches of three—turned up on black markets for historical art all over the world. It soon became clear, however, that many of these pieces were not the original ones but faked. For example, the first four of the size-three batches of mosaic pieces found were $S_1 = \{b_{17}, b_{3,471}, b_{4,946,071}\}$, $S_2 = \{b_{17}, b_{463}, b_{94,228}\}$, $S_3 = \{b_{231}, b_{56,463}, b_{12,094,578}\}$, and $S_4 = \{b_{17}, b_{3,471}, b_{94,228}\}$ —obviously, these three-element subsets of B weren’t disjoint and so some of them contained faked mosaic pieces. In total, a collection $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$ of $n = 55,557,390$ size-three batches of mosaic pieces (each S_i being offered for only one dollar) were spotted on black markets world-wide. “I do not care if they

are original or faked,” the bishop was quoted as saying, “I just want to have one (real or fake) copy of each piece of my mosaic. Look at my cathedral: It is *naked* inside!” So he started collecting money to buy enough size-three batches of mosaic pieces to be able to rebuild the complete mosaic in the cathedral, and eventually he had collected a total of exactly five million dollars. The problem he was facing now is the X3C problem: Is it possible to find five million sets S_i that exactly cover B ?⁸

It is not hard to believe that (large enough) instances of the X3C problem are computationally intractable, and it indeed is known that X3C is NP-complete (see, e.g., Garey and Johnson [GJ79]).

Turning back to the proof, let an instance $(B, \mathcal{S})$ of X3C be given, where $B = \{b_1, b_2, \dots, b_{3m}\}$, $m \geq 1$, $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$, and for each i , $1 \leq i \leq n$, we have that $S_i \subseteq B$ and $\|S_i\| = 3$. Define the value $h_j = \|\{S_i \in \mathcal{S} \mid b_j \in S_i\}\|$ for each j , $1 \leq j \leq 3m$. Define the election (C, V) , where $C = B \cup \{p, y, z\}$ is the set of candidates with the distinguished candidate p (i.e., p is the candidate the chair wishes to make the unique winner of the election), and where V is defined to consist of the following $2n + 2m + 3$ ballots:

1. For each i , $1 \leq i \leq n$, there is one voter who approves of the candidates in $\{z\} \cup S_i$ and who disapproves of the candidates in $\{p, y\} \cup (B - S_i)$.
2. For each i , $1 \leq i \leq n$, there is one voter who approves of the candidates in $\{p\} \cup B_i$, where $B_i = \{b_j \in B \mid i + h_j \leq n\}$, and who disapproves of the candidates in $\{y, z\} \cup (B - B_i)$.
3. There are $m + 1$ voters who each approve of only y and disapprove of the candidates in $\{p, z\} \cup B$.
4. There are $m + 2$ voters who each approve of the candidates in $\{p, z\} \cup B$ and disapprove of only y .

Note that $\text{score}_{(C,V)}(y) = m + 1$ and $\text{score}_{(C,V)}(p) = \text{score}_{(C,V)}(z) = \text{score}_{(C,V)}(b_j) = n + m + 2$ for all j , $1 \leq j \leq 3m$.

The following example illustrates the construction.

Example 3.2 Suppose we are given the X3C instance $(B, \mathcal{S})$, where $B = \{b_1, b_2, \dots, b_6\}$ and $\mathcal{S} = \{S_1, S_2, S_3\}$ with $S_1 = \{b_1, b_3, b_5\}$, $S_2 = \{b_2, b_4, b_6\}$, and $S_3 = \{b_1, b_4, b_6\}$. Our reduction then yields the instance (C, V, p) of AV-CCPV (in model TP) with candidate set $C = \{p, y, z\} \cup B$ and with V consisting of the 13 ballots shown in Table 4.

It is easy to see that this X3C instance has an exact cover: $S_1 \cup S_2 = B$. Our reduction has the property that p can be made the unique approval winner by partition of voters in model TP:

⁸This is a fictional example to illustrate X3C.

C	1st group	2nd group	3rd group	4th group	Score
p	0 0 0	1 1 1	0 0 0	1 1 1 1	7
y	0 0 0	0 0 0	1 1 1	0 0 0 0	3
z	1 1 1	0 0 0	0 0 0	1 1 1 1	7
b_1	1 0 1	1 0 0	0 0 0	1 1 1 1	7
b_2	0 1 0	1 1 0	0 0 0	1 1 1 1	7
b_3	1 0 0	1 1 0	0 0 0	1 1 1 1	7
b_4	0 1 1	1 0 0	0 0 0	1 1 1 1	7
b_5	1 0 0	1 1 0	0 0 0	1 1 1 1	7
b_6	0 1 1	1 0 0	0 0 0	1 1 1 1	7

Table 4: 13 ballots generated from a “yes” instance of X3C.

Subelection (C, V_1)				Score
C	1st	3rd		
p	0 0	0 0 0		0
y	0 0	1 1 1		3
z	1 1	0 0 0		2
b_1	1 0	0 0 0		1
b_2	0 1	0 0 0		1
b_3	1 0	0 0 0		1
b_4	0 1	0 0 0		1
b_5	1 0	0 0 0		1
b_6	0 1	0 0 0		1

Subelection (C, V_2)				Score
C	1st	2nd	4th	
p	0	1 1 1	1 1 1 1	7
y	0	0 0 0	0 0 0 0	0
z	1	0 0 0	1 1 1 1	5
b_1	1	1 0 0	1 1 1 1	6
b_2	0	1 1 0	1 1 1 1	6
b_3	0	1 1 0	1 1 1 1	6
b_4	1	1 0 0	1 1 1 1	6
b_5	0	1 1 0	1 1 1 1	6
b_6	1	1 0 0	1 1 1 1	6

Table 5: Subelections (C, V_1) and (C, V_2) obtained by voter partition from (C, V) in Example 3.2.

Partition V into (V_1, V_2) , where V_1 contains the two voters of the first group that correspond to the cover (these voters are given in boldface in Table 4) and all voters of the third group, and where V_2 contains the remaining voters. The resulting subelections are shown in Table 5. Since y is the unique approval winner of subelection (C, V_1) and p is the unique approval winner of subelection (C, V_2), they are the only candidates proceeding to the final round, where p beats y by 7 to 3.

Turning back to the proof of Theorem 3.5, we will now show that, in general, $\mathcal{S}$ has an exact

cover for B if and only if p can be made the unique approval winner by partition of voters in model TP.

From left to right: Suppose $\mathcal{S}$ contains an exact cover for B . Then the following partition of V into two subsets makes p the unique approval winner. Let V_1 contain the $m + 1$ voters of the third group (i.e., those voters approving of only y) and the m first-group voters corresponding to the cover, and let V_2 be $V - V_1$. It is easy to see that y is the unique approval winner of subelection (C, V_1) and p is the unique approval winner of subelection (C, V_2) . And when p and y meet in the second (and final) stage of the election, p will be the unique approval winner.

From right to left: Suppose p can be made the unique approval winner by partition of voters in model TP. So, in particular, p must win the final round. Recall, however, that everybody in $B \cup \{p, z\}$ has $n + m + 2$ approvals with respect to the whole electorate V , so each of the candidates in $B \cup \{z\}$ would tie for winner with p in the final round. Thus y is the only candidate who can run against p in the final round. Since we are in the TP model, it follows that each of the two subelections must have a unique approval winner: One must have y and the other one must have p .

Let (V_1, V_2) be such a partition of V where y is the unique approval winner of, say, subelection (C, V_1) , and p is the unique approval winner of the other subelection, (C, V_2) . Without loss of generality, we may assume that none of the voters in V_1 approve of p , so V_1 contains only voters of the first and the third group.

For p to become the unique approval winner of subelection (C, V_2) , p must in particular defeat each $b_j \in B$. For each $b_j \in B$, all voters of the third and fourth group approve of b_j exactly if they approve of p , so they don't help p to get an advantage over b_j and may thus be disregarded. If all the remaining voters (those of the first and second group) were in V_2 , each b_j would tie p in (C, V_2) . But since every $b_j \in B$ must lose at least one approval against p in (C, V_2) (and since all second-group voters approve of p and so, by our assumption, stay in V_2), the first-group voters that are not in V_2 (i.e., those in V_1) must form a cover for B .

However, since y is the unique approval winner of (C, V_1) and since y has $m + 1$ approvals overall, z can have no more than m approvals in (C, V_1) . However, among the voters in V_1 (to which, as noted above, only first- and third-group voters belong), z is approved by only first-group voters, so it can have no more than m approvals from these voters. Thus, V_1 contains no more than m (and thus exactly m) first-group voters, which represent an exact cover for B . $\square$

3.3.3 Adding and Deleting Voters

We have saved for last those cases whose motivations probably have the most emotional resonance for each of us, as voters. These cases are those related to adding and deleting voters.

We start by stating these control types.

Control by Adding Voters

Name: $\mathcal{E}$ -CCAV and $\mathcal{E}$ -DCAV.

Given: A set C of candidates, two disjoint collections of voters, V and W , represented via ballots over C , a distinguished candidate $p \in C$, and a nonnegative integer k . (V is sometimes called the registered voter set, and W , the pool of voters available for adding, is sometimes called the unregistered voter set.)

Question ($\mathcal{E}$ -CCAV): Is there a subset Q , $\|Q\| \leq k$, of voters in W such that the voters in $V \cup Q$ elect p as the unique winner according to system $\mathcal{E}$?

Question ($\mathcal{E}$ -DCAV): Is there a subset Q , $\|Q\| \leq k$, of voters in W such that the voters in $V \cup Q$ do not elect p as a unique winner according to system $\mathcal{E}$?

Control by Deleting Voters

Name: $\mathcal{E}$ -CCDV and $\mathcal{E}$ -DCDV.

Given: A set C of candidates, a collection V of voters represented via ballots over C , a distinguished candidate $p \in C$, and a nonnegative integer k .

Question ($\mathcal{E}$ -CCDV): Is it possible to by deleting at most k voters ensure that p is the unique winner of the resulting $\mathcal{E}$ election?

Question ($\mathcal{E}$ -DCDV): Is it possible to by deleting at most k voters ensure that p is not a unique winner of the resulting $\mathcal{E}$ election?

The reason we said above that these are the control types with the most emotional resonance is that these types loosely model activities that go to the heart of people's participation in elections. For example, deleting voters models voter suppression, where the chair's budget is enough to (by whatever means—visits, intimidation, spreading rumors, buying ads, making phone calls) keep any

choice of k voters from voting. Similarly, adding voters can be viewed as loosely modeling get-out-the-vote drives, e.g., the chair’s budget can pay to drive the chair’s choice of k people to the polls.

With respect to these control types, AV and SP-AV behave basically the same, and so it is not surprising that the results here are identical for AV and SP-AV: Both are vulnerable in the destructive cases and resistant in the constructive cases. As our final sample proof of this section, we prove the following result.

Theorem 3.6 ([ENR08b]) *SP-AV is vulnerable to destructive control by deleting voters and is resistant to constructive control by adding voters.*

Proof. In both cases, susceptibility immediately follows from the fact that the proof of susceptibility for destructive control by partition of voters from Theorem 3.4 also shows that SP-AV is susceptible to destructive control by partition of voters and the results of [HHR07a] that establish links between the susceptibility claims for certain types of control. In particular, it is known that every voiced voting system that is susceptible to destructive control by partition of voters in model TE or TP is also susceptible to destructive control by deleting voters [HHR07a, Theorem 4.3]. Susceptibility to destructive control by deleting voters in turn is equivalent to susceptibility to constructive control by adding voters [HHR07a, Theorem 4.1].

For concreteness, here is a quite simple example: Consider an election with two candidates, c and d , having two ballots of the form $c \mid d$ and one ballot of the form $d \mid c$. So c is the unique SP-AV winner. However, deleting one ballot of the form $c \mid d$ yields a tie between c and d . Thus SP-AV is susceptible to destructive control by deleting voters. On the other hand, if we view one ballot of the form $c \mid d$ as that of an unregistered voter, and the remaining two ballots, $c \mid d$ and $d \mid c$, as those of registered voters, then we can turn c from not being a unique SP-AV winner into the unique SP-AV winner by adding the originally unregistered ballot. Thus SP-AV is also susceptible to constructive control by adding voters.

To prove that SP-AV is vulnerable to destructive control by deleting voters, we present an algorithm for solving the problem SP-AV-DCDV in polynomial time. The algorithm takes as input an election (C, V) (that fulfills the requirements of SP-AV), a distinguished candidate p (whom the chair seeks to prevent from being a unique winner), and a nonnegative integer k (the maximum number of voters allowed to be removed from the election). The output of the algorithm either will be “control impossible” (if it is not possible to via deleting at most k votes prevent p from being a unique SP-AV winner), or it will be a subset $V' \subseteq V$ with $\|V'\| \leq k$ such that p is not a unique winner of the election $(C, V - V')$.

The algorithm first checks the following trivial cases:

1. If C contains p alone, then output “control impossible” and stop. (There is no other candidate who could possibly prevent p from being the unique SP-AV winner.)
2. Else if p already is not a unique SP-AV winner, then output the empty set as the set V' of voters to be deleted. (There is no need for the chair to intervene in this case.)

If none of the trivial cases applies, we know that $\|C\| \geq 2$ and p has more approvals than any other candidate in C . For each $c \in C$, let $\text{surplus}_{(C,V)}(p, c) = \text{score}_{(C,V)}(p) - \text{score}_{(C,V)}(c)$. Note that $\text{surplus}_{(C,V)}(p, c)$ is positive for each candidate $c \neq p$ in C . Now, the algorithm determines some candidate $q \neq p$ in C with smallest $\text{surplus}_{(C,V)}(p, q)$, and if $\text{surplus}_{(C,V)}(p, q) > k$ then it outputs “control impossible” and stops. In this case, deleting any choice of at most k voters will not dethrone p as the unique SP-AV winner. Otherwise (i.e., if $\text{surplus}_{(C,V)}(p, q) \leq k$), the algorithm outputs, as the set V' of voters to be deleted, $\text{surplus}_{(C,V)}(p, q)$ voters who approve of p and disapprove of q ,⁹ and stops.

To prove that SP-AV is resistant to constructive control by adding voters, we again give a reduction from the NP-complete problem X3C, which was defined in the proof of Theorem 3.5. Let an instance $(B, \mathcal{S})$ of X3C be given, where $B = \{b_1, b_2, \dots, b_{3m}\}$ (we assume $m > 1$, which is possible because the thus modified problem is still NP-complete), $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$, and $S_i \subseteq B$ with $\|S_i\| = 3$ for each i , $1 \leq i \leq n$.

Given $(B, \mathcal{S})$, construct an instance (C, V, W, p, m) of SP-AV-CCAV as follows. Let $C = B \cup \{p\}$, where p is the distinguished candidate. Let V contain $m - 2$ registered voters of the form $B \mid p$, and let W consist of the following n unregistered voters: For each i , $1 \leq i \leq n$, there is one voter of the form $p \mid (B - S_i)$ in W . Clearly, p is not a unique SP-AV winner of the election (C, V) , since $\text{score}_{(C,V)}(p) = 0$ and $\text{score}_{(C,V)}(b_j) = m - 2$ for each j , $1 \leq j \leq 3m$.¹⁰

Before proceeding with the proof, we present a small example to illustrate the construction.

Example 3.3 Consider the “yes” instance $(B, \mathcal{S})$ of X3C that is defined by $B = \{b_1, b_2, \dots, b_9\}$ (so $m = 3$) and $\mathcal{S} = \{S_1, S_2, S_3, S_4\}$ with $S_1 = \{b_1, b_3, b_5\}$, $S_2 = \{b_2, b_4, b_6\}$, $S_3 = \{b_1, b_4, b_6\}$, and $S_4 = \{b_7, b_8, b_9\}$. From this instance we construct the SP-AV-CCAV instance $(C, V, W, p, 3)$ with

⁹It is easy to see that, by definition of $\text{surplus}_{(C,V)}(p, q)$, such voters do exist.

¹⁰It is worth discussing the boundary case of $m = 2$ here, which is the smallest m possible. In this case, the election (C, V) has seven candidates, but none of these candidates is approved by any voter, simply because there are no voters (due to $\|V\| = m - 2 = 0$). By the definition of SP-AV winner (which, recall, is any candidate with the largest number of approvals), each of these seven candidates in C is a SP-AV winner, so it indeed is true that p is not a *unique* SP-AV winner of (C, V) , even if $m = 2$.

V: one registered voter										
b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9		p
W: four unregistered voters										
p	b_1	b_3	b_5		b_2	b_4	b_6	b_7	b_8	b_9
p	b_2	b_4	b_6		b_1	b_3	b_5	b_7	b_8	b_9
p	b_1	b_4	b_6		b_2	b_3	b_5	b_7	b_8	b_9
p	b_7	b_8	b_9		b_1	b_2	b_3	b_4	b_5	b_6
W': four unregistered voters										
p	b_1	b_3	b_5		b_2	b_4	b_6	b_7	b_8	b_9
p	b_2	b_4	b_6		b_1	b_3	b_5	b_7	b_8	b_9
p	b_1	b_4	b_6		b_2	b_3	b_5	b_7	b_8	b_9
p	b_2	b_5	b_8		b_1	b_3	b_4	b_6	b_7	b_9

Table 6: Registered and unregistered voters generated from a “yes” and from a “no” instance of X3C.

candidate set $C = \{p\} \cup B$, distinguished candidate p , and with registered and unregistered voters, V and W , as shown in Table 6.

Clearly, p is not a unique SP-AV winner of the election (C, V) , as p has zero approvals and each $b_j \in B$ has one approval. However, adding to V the first two and the last of the unregistered voters of W (which correspond to an exact cover for B) makes p the unique SP-AV winner of the resulting election, as in this election p has three approvals, but each $b_j \in B$ has only two.

For the sake of contrast, consider the “no” instance $(B, \mathcal{S}')$ of X3C that is obtained from $(B, \mathcal{S})$ by modifying only the fourth set in $\mathcal{S}$ from S_4 to $S'_4 = \{b_2, b_5, b_8\}$, so $\mathcal{S}' = \{S_1, S_2, S_3, S'_4\}$. Now, this instance is transformed by our reduction into the SP-AV-CCAV instance $(C, V, W', p, 3)$, where W' can also be found in Table 6. Clearly, $(C, V, W', p, 3)$ is a “no” instance of SP-AV-CCAV, since adding any subset W'' of at most three voters from W' to V will fail to make p a unique SP-AV winner of the election $(C, V \cup W'')$. For example, if we added the first two and the last of the unregistered voters of W' to V then p would tie for winner with both b_2 and b_5 , with each having three approvals.

We now prove that $\mathcal{S}$ has an exact cover for B if and only if p can be made the unique SP-AV winner by adding at most m voters.

If $\mathcal{S}$ contains an exact cover for B , then let $W' \subseteq W$ be the set of voters corresponding to this cover and add W' to V . Since $\text{score}_{(C, V \cup W')}(p) = m$ and $\text{score}_{(C, V \cup W')}(b_j) = m - 1$ for each j , $1 \leq j \leq 3m$, p is the unique SP-AV winner of the election $(C, V \cup W')$.

Conversely, suppose that p can be made the unique SP-AV winner by adding at most m voters from W to V . Every candidate b_j in B has $m - 2$ approvals more than p in (C, V) , and adding any

voter from W will give both p and three members of B one more approval. Thus for p to become the unique SP-AV winner we need to add exactly m voters from W , while making sure that no candidate from B gains more than one additional approval. It follows that the m voters added correspond to an exact cover for B . $\square$

3.4 Conclusions

Table 1 already summarized the control results. However, some comments are in order. We mention that Table 1 reflects the fact that each of the 44 cases' complexity is known. None of the cases remain open. We also note that SP-AV (see however the discussion and caveats of Section 2) has a very large number of resistances. This is clearly a strong point in its favor, although of course one's choice of an election system for any particular task will depend on the task, and maximizing control resistances will rarely be the only—or even the most important—factor one weighs in choosing one's system. Other factors may include simplicity (for the voter), perceived fairness, acceptability to the electorate, resistance to voter manipulation (see [BTT92, FHHR09]), etc. And even if control resistances were what one cared about, the Llull/Copeland complex of systems, while having fewer resistances in number, has some resistances that SP-AV lacks, and so again one would judge by what resistances are most needed for the task at hand—what attacks we most want to be protected from. Nonetheless, it is clear that SP-AV is an interesting system having a quite large number of control resistances, and as such is worth being at least seriously considered—weighing its overall strengths and weaknesses—when one is choosing an election system for a task.

4 The Complexity of Bribery for Approval Voting

In this section, we are concerned with the complexity of bribery for approval voting. The notion of bribery was introduced by Faliszewski, Hemaspaandra, and Hemaspaandra [FHH06a, FHH06b], who studied the hardness of bribery for voting systems as diverse as approval voting, scoring protocols (an important class of voting systems including in particular Borda count, plurality, veto, and k -approval), and Dodgson voting ([Dod76], see [HHR97, HH06, HH, CCF⁺09] for results on the computational aspects of Dodgson's system). Bribery has been subsequently explored for a variety of other voting systems (see, e.g., [FHHR07, FHHR08a, FHHR08b, Fal08]). Here we will focus on the complexity results for bribery with respect to approval voting established by Faliszewski et al. [FHH06a, FHH06b].

As mentioned in the Introduction, bribery models scenarios in which an external agent, the “briber,” seeks to make his or her favorite candidate win (i.e., we in this section focus on the constructive case only) by bribing some voters to change their votes. As such, bribery can be seen as sharing some aspects with control (namely, that an external actor seeks to change the electoral outcome) and some with manipulation (namely, that the voters’ ballots may be changed). Unlike the chair in control settings, the briber doesn’t alter the procedure of an election but rather alters the voters’ ballots, and unlike the strategic voters in manipulation settings, it here is the briber (and not some of the voters) who seeks to do something bad.

For each of the bribery problems we will define, the briber’s budget will be limited and we will consider the following variants of how the budget can be spent in order to reach the briber’s goal:

1. Voters can be “weighted” or “unweighted.” For example, if a voter of weight three approves of some candidate then this candidate walks off with the equivalent of three weight-one approvals from just this voter. Weighted voting occurs in many real-life settings. In voting on referenda, stockholders’ votes are typically weighted by the number of shares they own.
2. Voters may or may not have price tags. The priced-voters case again is modeling a natural situation, namely, it reflects the fact that some voters may be more expensive to bribe than others.
3. For both the weighted-voters case and the priced-voters case, one can distinguish between whether weights/prices are represented in binary or in unary. Clearly, for the computational complexity of the problems thus defined the representation of the problem instances can—and we will show that in some cases it does—make a difference.
4. Finally, in addition to bribery where a voter’s ballot either is or is not bought by the briber, we will also consider a more fine-tuned, local approach, called “microbribery,” where the briber pays for each bit-flip.¹¹ This model reflects the fact that voters naturally may have stronger feelings about some candidates than about others, and so a voter’s approval or disapproval of some candidate may be for sale but the same voter may not be willing to change his or her approval/disapproval of some other candidate. If it comes to convictions, voters may have preferences that money can’t buy.

¹¹The study of microbribery was initiated in [FHH06a], where it however was notated bribery’, and was also studied in [FHHR07]. The term “microbribery” was first used for this concept in [FHHR08a], which studies (among other things) microbribery of the Llull/Copeland election systems.

We start by formally defining the most basic variant of bribery for approval voting. Unlike in Section 3, we define all bribery problems in the nonunique-winner model, since that is the core model taken by Faliszewski et al. [FHH06a,FHH06b] (although do note that essentially all their results also hold in the nonunique winner mode) and the other papers on bribery, and it makes sense to keep the studies of a given type of attack as uniform as possible in their model. We mention in passing that some papers study bribery, manipulation, or control in both the unique-winner and the nonunique-winner models. For example, Faliszewski et al. [FHHR08b] do so for bribery and control regarding the Llull/Copeland complex of systems, and they obtain the same complexity results for both models in each case. Similarly, Hemaspaandra, Hemaspaandra, and Rothe [HHR06] prove that the complexity of winner determination, which originally was shown in the nonunique-winner model for Dodgson [HHR97], Young [RSV03], and Kemeny elections [HSV05], remains the same in the unique-winner model. To the best of our knowledge, the only complexity results where the unique-winner model parts company with the nonunique-winner model are due to Faliszewski, Hemaspaandra, and Schnoor [FHS08] and are related to manipulating Copeland elections.

Name: AV-BRIBERY.

Given: A set C of candidates, a collection V of voters represented via ballots over C , a distinguished candidate p , and a nonnegative integer k (the “budget”).

Question: Is it possible to change at most k voters’ ballots so that p is an approval winner of the resulting election?

In an AV-BRIBERY instance, all voters are both unweighted and unpriced. The corresponding bribery problem for weighted but unpriced voters is denoted by AV-WEIGHTED-BRIBERY, that for unweighted but priced voters by AV-\$BRIBERY, and if the voters both are weighted and have price tags, we write AV-WEIGHTED-\$BRIBERY.

The first result in this section is that even AV-BRIBERY, the simplest of these four problems, is NP-complete. Since this problem is a special case of each of the other three problems just defined, they each immediately inherit the NP-hardness lower bound of AV-BRIBERY. On the other hand, it is easy to see that each of these four problems is contained in NP: Given an instance (C, V, p, k) , nondeterministically guess a bribery action involving at most k voters and verify deterministically that p wins the resulting election. Obviously, this can be done in polynomial time, even if weights and prices are given in binary.¹²

¹²Clearly, representing weights or prices in unary provides a less succinct input than using the binary representation.

Theorem 4.1 ([FHH06a,FHH06b]) AV-BRIBERY is NP-complete.

Proof. Membership in NP was justified above. It remains to prove that AV-BRIBERY is NP-hard. To this end, we now describe a reduction from the NP-complete problem X3C (which was defined in the proof of Theorem 3.5). Let an instance $(B, \mathcal{S})$ of X3C be given, where $B = \{b_1, b_2, \dots, b_{3m}\}$, $m \geq 1$, $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$, and for each i , $1 \leq i \leq n$, we have that $S_i \subseteq B$ and $\|S_i\| = 3$. Without loss of generality, we may assume that $n \geq m$, since otherwise $\mathcal{S}$ wouldn't have an exact cover for B , and mapping such a “no” instance of X3C to some fixed “no” instance of AV-BRIBERY would correctly handle this case. We again need the values h_j , which were defined in the proof of Theorem 3.5 as $h_j = \|\{S_i \in \mathcal{S} \mid b_j \in S_i\}\|$ for each j , $1 \leq j \leq 3m$.

Define the election (C, V) , where $C = B \cup \{p\}$ is the set of candidates with the distinguished candidate p , and where V is defined to consist of the following $(3m + 2)n + 2m - \sum_{j=1}^{3m} h_j = 3mn + 2m - n$ ballots:

1. For each i , $1 \leq i \leq n$, there is one voter who approves of the three candidates in S_i and who disapproves of all other candidates.
2. For each j , $1 \leq j \leq 3m$, there are $n - h_j + 1$ voters who each approve of only b_j and who disapprove of all other candidates.
3. There are $n - m$ voters who each approve of only p and who disapprove of the candidates in B .

Note that $\text{score}_{(C,V)}(p) = n - m$ and $\text{score}_{(C,V)}(b_j) = n + 1$ for each j , $1 \leq j \leq 3m$.

Define our AV-BRIBERY instance to be (C, V, p, m) . As in the previous reductions presented in Section 3, we give an example to illustrate this construction of (C, V, p, m) from the given X3C instance $(B, \mathcal{S})$.

Example 4.1 Suppose we are given the X3C instance $(B, \mathcal{S})$, where $B = \{b_1, b_2, \dots, b_6\}$ and $\mathcal{S} = \{S_1, S_2, S_3, S_4\}$ with $S_1 = \{b_1, b_2, b_3\}$, $S_2 = \{b_2, b_4, b_6\}$, $S_3 = \{b_1, b_3, b_5\}$, and $S_4 = \{b_1, b_2, b_6\}$. Our reduction then yields the instance $(C, V, p, 2)$ of AV-BRIBERY with candidate set $C = \{p\} \cup B$ and with V consisting of the 24 ballots shown in Table 7.

Since $S_2 \cup S_3 = B$, $\mathcal{S}$ has an exact cover for B . Our reduction ensures that p can be made an approval winner by bribing the two voters of the first group that correspond to this exact cover (i.e., the two boldfaced first-group voters in Table 7): If their ballots are changed such that they both

This implies that a problem using unary encoding is at most as hard, computationally, as the corresponding problem using binary encoding.

C	1st				2nd																		3rd		Score																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
p	0	0	 

Table 7: 24 ballots generated from a “yes” instance of X3C.

approve of only p and disapprove of all candidates in B , then all candidates are winners with score 4 in the resulting election.

We now prove that $\mathcal{S}$ has an exact cover for B if and only if p can be made an approval winner by bribing at most m voters.

From left to right: Suppose that $\mathcal{S}$ has an exact cover for B . Simply bribe the voters corresponding to this exact cover by changing their ballots such that each bribed voter approves of p and disapproves of the candidates in B . Then p gains m approvals (i.e., $\text{score}_{(C,V')}(p) = n$, where V' denotes the collection of ballots after the bribery) and every $b_j \in B$ loses one approval (i.e., $\text{score}_{(C,V')}(b_j) = n$). Thus p is an approval winner in (C, V') .

From right to left: Suppose that p can be made an approval winner by bribing at most m voters. Let (C, V') denote the election that results from any such bribery. By bribing at most m voters, p can gain no more than m additional approvals. Thus $\text{score}_{(C,V')}(p) \leq (n - m) + m = n$. On the other hand, all candidates in B have $n + 1$ approvals initially, so for p to become an approval winner by bribing at most m voters, each $b_j \in B$ has to lose at least one approval by the bribery. It follows that exactly m voters must have been bribed to yield (C, V') , and these m voters correspond to an exact cover for B . $\square$

Corollary 4.2 ([FHH06a, FHH06b]) *Each of the problems AV-WEIGHTED-BRIBERY, AV-\$BRIBERY, and AV-WEIGHTED-\$BRIBERY is NP-complete.*

We now turn to the formal definition of the microbribery problem for approval voting. Again, we start with the most basic case where voters are unweighted and don’t have price tags.

Name: AV-MICROBRIBERY.

Given: A set C of candidates, a collection V of voters represented via ballots over C , a distinguished candidate p , and a nonnegative integer k .

Question: Is it possible to flip at most k entries in the ballots (to be clear: not k entries per ballot but k entries overall, in all the ballots) so that p is an approval winner of the resulting election?

As with bribery, AV-WEIGHTED-MICROBRIBERY denotes the microbribery problem for weighted but unpriced voters within approval voting, AV-\$MICROBRIBERY denotes this problem for unweighted but priced voters (each bit flip may have a different price), and we write AV-WEIGHTED-\$MICROBRIBERY if the voters both are weighted and have price tags. Here, it will also make sense to distinguish between binary and unary representations for weights or prices. The names of the above-defined four microbribery problems for AV refer to the more succinct binary representation of, respectively, weights and prices, which is the standard way of representing numbers. To indicate that either weights or prices or both are given in unary, we use the subscript “unary” at the corresponding place in the problem name. For example, AV-WEIGHTED_{unary}-\$MICROBRIBERY denotes the same problem as AV-WEIGHTED-\$MICROBRIBERY but with weights given in unary and prices given in binary. It is known that the microbribery problem for AV is easy to solve even if both weights and prices are given, provided that at least one of them is represented in unary. We omit the proof.

Theorem 4.3 ([FHH06a,FHH06b]) AV-MICROBRIBERY, AV-WEIGHTED_{unary}-\$MICROBRIBERY_{unary}, AV-WEIGHTED_{unary}-\$MICROBRIBERY, and AV-WEIGHTED-\$MICROBRIBERY_{unary} can each be solved in polynomial time.

In contrast, if both weights and prices are encoded in binary, the microbribery problem for AV is NP-complete.

Theorem 4.4 ([FHH06a,FHH06b]) AV-WEIGHTED-\$MICROBRIBERY is NP-complete.

Proof. Again, it is clear that AV-WEIGHTED-\$MICROBRIBERY is in NP. Thus, it remains to prove that AV-WEIGHTED-\$MICROBRIBERY is NP-hard. To this end, we reduce from the following famous problem:

Name: PARTITION.

Given: A nonempty sequence $s_1, s_2, \dots, s_n$ of positive integers such that $\sum_{i=1}^n s_i$ is an even number.

Question: Does there exist a subset $A \subseteq \{1, 2, \dots, n\}$ such that $\sum_{i \in A} s_i = \sum_{i \in \{1, 2, \dots, n\} - A} s_i$?

Before we proceed with the proof, let us illustrate this problem by an example that continues the example for the X3C problem presented in the proof of Theorem 3.5. Recall from that previous example that a magnificent mosaic had been stolen from the Cathedral of Monreale on Sicily and then was sold—piece by piece, along with faked duplicates, and always in batches of three—on black markets world-wide. Now, Giuseppe Greco and Salvatore Inzerillo, who had planned, organized, and executed this coup, were meeting in a bar in the port district of Palermo to divide what they had earned, namely \$55,557,390. To hide their tracks, they had changed this amount of money into 1,235 bars of gold with different sizes, $s_1, s_2, \dots, s_{1,235}$, each s_i being the integer amount of dollars the i th bar of gold was worth.

“Now let me get this straight,” said Giuseppe who knew Salvatore all too well from other deals. “This time we do it fifty-fifty!”

“Yes, fine with me,” replied Salvatore craftily. “So tell me, my friend: Which of the gold bars are for me?”

Giuseppe set down and started thinking. He was counting the gold bars, playing with them, weighing them in his hands, and comparing their dollar amounts. He was scratching his head. Meanwhile, Salvatore was watching him silently. Giuseppe was thinking for more than an hour. Then he said: “No! You tell me! Which of the gold bars are mine?”

“Well,” said Salvatore slowly, “all I know about this, and these were my father’s last words before he died, so it must be true: This is a very hard problem, and you cannot solve it in a trice. So it just won’t be possible for us to make it fifty-fifty. You can try to solve this problem by brute force, but this won’t be possible in a reasonable amount of time!”

Salvatore was right in one regard: PARTITION is an NP-complete problem (see, e.g., Garey and Johnson [GJ79]) and thus, indeed, is very hard to solve. But he wasn’t right in another regard: Giuseppe did solve the problem he was facing by brute force before dawn. Salvatore was found the next day, floating face-down in the Mediterranean near the port of Palermo. And Giuseppe and all the gold were gone.

Now, back to the proof, let an instance $(s_1, s_2, \dots, s_n)$ of PARTITION be given, where we have $\sum_{i=1}^n s_i = 2S$ and where S and each s_i is a positive integer. Define the election (C, V) , where $C = \{p, x\}$ is the set of candidates with p being the distinguished candidate, and where V consists of the following $n + 1$ ballots:

1. There is one voter of weight S who approves of p and disapproves of x , and the price for flipping any entry in this voter’s ballot is $2S + 1$.

V	Weight	Price		Ballot	
		p	x	p	x
v_0	10	21	21	1	0
v_1	9	9	21	0	1
v_2	2	2	21	0	1
v_3	3	3	21	0	1
v_4	4	4	21	0	1
v_5	1	1	21	0	1
v_6	1	1	21	0	1
Score				10	20

Table 8: Seven ballots generated from a “yes” instance of PARTITION.

- For each i , $1 \leq i \leq n$, there is one voter of weight s_i who approves of x and disapproves of p , where the price for flipping this voter’s p -entry is s_i and the price for flipping this voter’s x -entry is $2S + 1$.

Our AV-WEIGHTED-\$MICROBRIBERY instance then is (C, V, p, S) . Note that $\text{score}_{(C,V)}(p) = S$ and $\text{score}_{(C,V)}(x) = 2S$. Thus x is the unique approval winner of the election (C, V) .

The following example illustrates the construction.

Example 4.2 Suppose we are given the PARTITION instance $(s_1, s_2, \dots, s_6)$, where $s_1 = 9$, $s_2 = 2$, $s_3 = 3$, $s_4 = 4$, $s_5 = 1$, and $s_6 = 1$. This is a “yes” instance of PARTITION, and a possible partition of $\{1, 2, \dots, 6\}$ is $\{1, 5\}$ and $\{2, 3, 4, 6\}$, since $s_1 + s_5 = 10 = s_2 + s_3 + s_4 + s_6$.

Our reduction then yields the instance $(C, V, p, 10)$ of AV-WEIGHTED-\$MICROBRIBERY with candidate set $C = \{p, x\}$ and with V consisting of the seven ballots shown in Table 8.

Clearly, x is the unique winner of (C, V) , as x has 20 approvals and p has only 10. However, p can be made an approval winner by microbribing the two voters that correspond to the $\{1, 5\}$ part of the partition given above. Namely, flipping the p -entries in the ballots of voters v_1 and v_5 costs exactly 10 units of bribery currency (which is just as much as the briber can afford). After the bribery, the voters v_1 and v_5 approve of both p and x , so both candidates, p and x , are approval winners with score 20.

Turning back to the proof, we now show that, in general, there is a set $A \subseteq \{1, 2, \dots, n\}$ such that $\sum_{i \in A} s_i = S$ if and only if p can be made an approval winner by a microbribery of cost at most S .

From left to right: Suppose there is a set $A \subseteq \{1, 2, \dots, n\}$ such that $\sum_{i \in A} s_i = S$. Change the disapprovals of p into approvals in the ballots of the voters corresponding to A . Clearly, this will cost $\sum_{i \in A} s_i = S$, and after that both p and x are approval winners with score $2S$.

From right to left: Suppose p can be made an approval winner by a microbribery of cost at most S . The given amount S allows only flips from 0 to 1 in the p -entries of the second voter group. Thus p can gain (at most) S additional approvals, whereas x 's score will still be $2S$ after the microbribery. Since for the voters whose ballots were changed to make p an approval winner, the weights and the costs for flipping their p -entries are the same, p must have gained exactly S approvals and the weights of these voters must also be exactly S . Thus there is a set $A \subseteq \{1, 2, \dots, n\}$ such that $\sum_{i \in A} s_i = S$. $\square$

5 Local Search Heuristics for Minimax Approval Voting

5.1 Minisum and Minimax Approval Voting

In the present section, we will focus on the complexity of electing a committee of fixed size via another variant of approval voting. (For more on the complexity issues in electing committees, we commend to the reader the work of Meir et al. [MPRZ08].)

The standard way of evaluating an approval election to obtain a committee of k candidates is to sum up the approvals for each candidate and to select k candidates with the highest number of approvals (where some tie-breaking rule can be used when there is more than one such size- k committee). This method is called the *minisum* procedure, since the outcome minimizes the sum of the Hamming distances of the winning committee to all ballots. Recall that the Hamming distance between two binary vectors is the minimum number of bit-flips needed to transform one vector into the other.

Brams, Kilgour, and Sanver [BKS04] (see also [BKS07a, BKS07b]) proposed a new evaluation method for approval elections to determine committees of fixed size, which is called the *minimax* procedure, since it minimizes the maximum of the Hamming distances between the winning committee and the ballots. The minisum procedure seeks to find an outcome that is close to many ballots, i.e., it minimizes the *total* dissatisfaction of the electorate, whereas the minimax procedure seeks to minimize the dissatisfaction of the most dissatisfied voter, even if this results in a higher total (or, equivalently, average) dissatisfaction.

Example 5.1 Table 9 gives an example of an election that aims to find a committee of size $k = 2$ by approval voting. There are four candidates and eight voters in this election, where three voters

Committee	Ballots			Evaluation	
	(3×) 1110	(2×) 0101	(3×) 1010	sum	max
1100	3	4	6	13	6
1010	3	8	0	11	8
1001	9	4	6	19	9
0110	3	4	6	13	6
0101	9	0	12	21	12
0011	9	4	6	19	9

Table 9: Electing a committee of size two via minisum and minimax evaluation for approval voting.

approve of the first three candidates, two voters approve of the second and the fourth candidate, and three voters approve of the first and the third candidate. In the first column, the possible committees with two candidates each are listed. The next three columns give the Hamming distances between each such committee and each of the three distinct ballots times their multiplicities, the “sum” column gives the sum of these values, and the “max” column gives their maximum.

A committee with a smallest entry in the “sum” column is a minisum (winner) committee, and in this example we happen to have a unique minisum committee, namely 1010. The two voters with ballots 0101, however, will be completely dissatisfied with the minisum outcome, since none of their candidates is in the minisum committee and all candidates they disapprove of are in this committee.

A committee with a smallest entry in the “max” column is a minimax (winner) committee, and in this example there are two minimax committees, 1100 and 0110.

Both the minisum and the minimax evaluation presented in Example 5.1 take only the number of identical ballots into account. A different approach, also proposed by Brams, Kilgour, and Sanver [BKS07a], is to base these evaluations on “proximity weights.” Instead of merely counting how often each ballot occurs, the “proximity” of any ballot to all other ballots is thus taken into account. Formally, the *proximity weight* of a given ballot v_i is defined by

$$(4) \quad \frac{m_i}{\sum_{j=1}^t m_j \cdot H(v_i, v_j)},$$

where $t > 1$ is the total number of distinct ballots (say, $v_1, v_2, \dots, v_t$ with $v_i \neq v_j$ for $i \neq j$), m_i is the multiplicity of v_i , and $H(v_i, v_j)$ is the Hamming distance between ballots v_i and v_j . Note that, to ensure that the denominator in (4) isn’t zero, we assume that there are at least two distinct ballots.

Committee	Ballots with proximity weights			Evaluation	
	$(3 \times) 1110$ $\frac{1}{3}$	$(2 \times) 0101$ $\frac{2}{21}$	$(3 \times) 1010$ $\frac{3}{11}$	sum	max
1100	693	264	1134	2091	1134
1010	693	528	0	1221	693
1001	2079	264	1134	3477	2079
0110	693	264	1134	2091	1134
0101	2079	0	2268	4347	2268
0011	2079	264	1134	3477	2079

Table 10: Electing a size-two committee via minisum and minimax approval voting with proximity weights.

Example 5.2 Table 10 shows the same election as in Example 5.1, except that the evaluation is made by calculating the Hamming distance of each committee to each distinct ballot times this ballot's proximity weight (instead of multiplying this Hamming distance with this ballot's multiplicity). To avoid fractions, all results are multiplied by the least common multiple of the distinct voters' denominators in (4), which is 693 in this example. This is reasonable, since the ratios between the single alternative committees remain the same, see Kilgour, Brams, and Sanver [KBS06].

Both the winning committee obtained via the minisum evaluation with proximity weights and that obtained via the minimax evaluation with proximity weights happen to coincide with the minisum committee calculated without proximity weights in Example 5.1. However, Brams, Kilgour, and Sanver [BKS07a] mention that minisum outcomes not based on proximity weights and minimax outcomes based on proximity weights can diverge maximally, i.e., all bits in the minisum outcome are flipped in the minimax outcome.

5.2 NP-Hardness and Approximability of Fixed-Size Minimax Approval Voting

Given an election with m candidates, n ballots, and a fixed committee size of k , the minisum evaluation (with or without proximity weights) is easy and can be done in polynomial time. If proximity weights are not considered, simply add the number of ones in all ballots for each candidate, and declare k candidates with the highest approval scores as winners to join the committee. If ties occur (e.g., if there are more than k candidates each having the highest approval score), the winning committee of size k may be selected among the possible size- k minisum committees by using some

fixed, computationally simple tie-breaking rule. If proximity weights are to be considered in a minimax evaluation, the algorithm is similarly efficient, since all proximity weights involved can be determined in polynomial time.

In a minimax evaluation, however, all $\binom{m}{k}$ committees in concept may need be considered, and although for each committee the Hamming distances to all ballots can be computed easily, this may be just too many committees. Indeed, LeGrand [LeG04] showed that the following problem is NP-complete.

Name: FIXED-SIZE MINIMAX APPROVAL VOTING (FSM-AV, for short).

Given: A set C of candidates, a collection V of voters represented via ballots over C , a nonnegative integer $k \leq \|C\|$, and a nonnegative integer $d \leq \|C\|$.

Question: Does there exist a vector $v \in \{0, 1\}^{\|C\|}$ such that v has exactly k ones and $H(v, v_i) \leq d$ for all $v_i \in V$, i.e., is there a committee of size k whose maximum Hamming distance to the ballots in V is at most d ?

Theorem 5.1 ([LeG04]) FSM-AV is NP-complete.

Proof. FSM-AV is easily seen to be in NP, so it remains to prove FSM-AV NP-hardness. LeGrand [LeG04] gave a reduction to FSM-AV from the NP-complete problem VERTEX COVER (see, e.g., Garey and Johnson [GJ79]), which is formally defined as follows.

Name: VERTEX COVER.

Given: An undirected graph G with vertex set $V(G) = \{v_1, v_2, \dots, v_n\}$ and edge set $E(G)$, and a nonnegative integer $k \leq n$.

Question: Does G have a vertex cover of size at most k , i.e., does there exist a subset $V' \subseteq V(G)$ with $\|V'\| \leq k$ such that for each edge $\{x, y\} \in E(G)$, we have that $\{x, y\} \cap V' \neq \emptyset$?

Does this problem look familiar to you? If it does, you've been carefully reading the proofs in Section 3! VERTEX COVER may be viewed as a special case of the problem HITTING SET, which was defined in the proof of Theorem 3.3. In fact, HITTING SET is nothing other than the vertex cover problem for hypergraphs (note: the hyperedges of a hypergraph may involve more than two vertices). So, any given instance (G, k) of VERTEX COVER may be viewed as the instance $(V(G), E(G), k)$ of HITTING SET. To see this, recall the intuitive explanation of HITTING SET from the proof of Theorem 3.3: Students signing up for courses and professors seeking to find a hitting set

Figure 1: Graph G for Example 5.3: Reducing VERTEX COVER to FSM-AV.

of size at most k among the students such that each course contains at least one hitting-set student. In this interpretation, VERTEX COVER is the special case of HITTING SET in which every course is attended by exactly two students, and if we draw a graph the vertices of which are the students and that has an edge between two vertices exactly if the corresponding students are jointly attending the same course, then determining a vertex cover of size at most k in the graph is just the same as asking for a hitting set of size at most k . (This observation immediately provides a reduction from VERTEX COVER to HITTING SET.)

We now describe a reduction from VERTEX COVER to FSM-AV.¹³ Given an instance (G, k) of VERTEX COVER, we will define an instance (C, V, k, k) of FSM-AV such that (C, V, k, k) is a “yes” instance of FSM-AV if and only if (G, k) is a “yes” instance of VERTEX COVER. Note that the maximum size of the vertex cover allowed, k , equals both the size of the committee represented by v and the maximum Hamming distance of v to the ballots in V .

Every vertex in G represents one candidate, so $C = \{c_i \mid v_i \in V(G)\}$, and every edge $\{v_i, v_j\}$ of G represents one voter who approves of the two candidates c_i and c_j and who disapproves of all other candidates. Formally, $V = \{b_{i,j} \mid \{v_i, v_j\} \in E(G)\}$, where $b_{i,j}$ is the bit-vector of length $\|C\|$ that has a one at positions i and j and a zero at all other positions. Obviously, the instance (C, V, k, k) of FSM-AV can be computed in time polynomial in the size of the given VERTEX COVER instance (G, k) .

Before proving the correctness of the reduction, let us illustrate the construction by an example.

¹³Note that the reduction presented here is based on (but is much simpler than) the reduction given by LeGrand [LeG04].

Edge $\{v_i, v_j\}$	$\{1, 3\}$	$\{2, 4\}$	$\{2, 6\}$	$\{3, 5\}$	$\{3, 6\}$	$\{4, 5\}$	$\{4, 6\}$	$\{5, 6\}$
Ballot $b_{i,j} \in V$	101000	010100	010001	001010	001001	000110	000101	000011
$H(001101, b_{i,j})$	3	3	3	3	1	3	1	3

Table 11: Eight ballots constructed from graph G and their Hamming distance to $v = 001101$.

Example 5.3 Consider the graph G with six vertices and eight edges that is shown in Figure 1. Since the edges $\{1, 3\}$, $\{4, 5\}$, and $\{2, 6\}$ have no vertex in common, each vertex cover of G must contain at least three vertices to cover these edges. One possible vertex cover of size three for G is the set $V' = \{3, 4, 6\}$, whose elements are in boldface in Figure 1. Thus $(G, 3)$ is a “yes” instance, but $(G, 2)$ is a “no” instance of VERTEX COVER.

Given the “yes” instance $(G, 3)$ of VERTEX COVER, our reduction yields the instance $(C, V, 3, 3)$ of FSM-AV, where $C = \{c_1, c_2, \dots, c_6\}$ is the set of candidates and V consists of the eight ballots shown in Table 11, where each ballot is a 6-bit binary vector that corresponds to one of the edges of G . Note that the vector $v = 001101$, which represents a committee, corresponds to the vertex cover $V' = \{3, 4, 6\}$, and the last row in Table 11 shows that the Hamming distance between v and all the ballots in V is at most three. Thus $(C, V, 3, 3)$ is a “yes” instance of FSM-AV, as desired.

However, decrementing the parameter k to 2 gives the “no” instance $(G, 2)$ of VERTEX COVER. Our reduction then yields the FSM-AV instance $(C, V, 2, 2)$, where the election (C, V) is as described above. One can easily check that no 6-bit vector with exactly two ones has a Hamming distance of at most two to all the ballots in V . So $(C, V, 2, 2)$ is a “no” instance of FSM-AV, as desired.

Now, returning to the proof of Theorem 5.1, we claim that G has a vertex cover of size k if and only if there is a vector $v \in \{0, 1\}^{\|C\|}$ such that v has exactly k ones and $H(v, b_{i,j}) \leq k$ for all $b_{i,j} \in V$.

From left to right: Suppose there is a vertex cover V' of size exactly k in G . Define the vector $v \in \{0, 1\}^{\|C\|}$ to have a one at position i if $v_i \in V'$, and to have a zero at position i otherwise. By construction, there are k ones in v , and since V' is a vertex cover of G , every ballot in V has at least one of its two ones at a position that is set to one in v . Thus the Hamming distance of v to each ballot $b_{i,j}$ in V is at most k , since v can be transformed into $b_{i,j}$ by flipping at most one zero in v to one and at most $k - 1$ ones in v to zeros.

From right to left: Suppose there is a vector $v \in \{0, 1\}^{\|C\|}$ such that v has exactly k ones and $H(v, b_{i,j}) \leq k$ for all $b_{i,j} \in V$. Define the set V' to contain exactly those vertices v_i for which v has a one at position i . Clearly, $\|V'\| = k$. Since $H(v, b_{i,j}) \leq k$, each ballot $b_{i,j} \in V$ must have at least

one of its two ones at a position at which also v has a one. Thus $V' \cap \{v_i, v_j\} \neq \emptyset$ for all edges $\{v_i, v_j\} \in E(G)$. That is, V' is a vertex cover of size k for G . $\square$

If an important problem turns out to be NP-hard, this doesn't make the problem go away: We still want to solve it, at least as well as we can. So, how can we cope with NP-hardness? There are several answers to this question; let us sketch some of them below.

In practical applications, we may content ourselves with an efficient heuristic algorithm that isn't *always* correct but that does work correctly for those inputs that typically occur in practice. For example, Homan and Hemaspaandra [HH] proposed an efficient greedy heuristic for finding Dodgson winners, and they proved that under the model of voting that political scientists call impartial culture this heuristic is a "frequently self-knowingly correct algorithm" (see also the closely related work of McCabe-Dansted, Pritchard, and Slinko [MPS08]; see also the discussion in Erdélyi et al. [EHR08, EHR07], which in turn discusses notions proposed by Procaccia and Rosenschein [PR07]). Regarding this approach to coping with NP-hardness, Section 5.3 will present some local search heuristics for minimax approval voting.

From a practical point of view, we may also be interested in solving NP-hard problems only for certain small parameters. For example, if the size of the allowed committee in FSM-AV instances is always bounded by a constant k then all $\binom{m}{k'} = \mathcal{O}(m^k)$, $k' \leq k$, possible committees where k' is the committee size can be evaluated in polynomial time, and if this constant bound on committee size is small then this naive algorithm may even be useful in practice. Similarly, if the number of voters in FSM-AV instances is bounded by a small constant, then the problem can be solved in polynomial time by an integer linear program with a constant number of constraints (see LeGrand, Markakis, and Mehta [LMM06, LMM07]). Such results fall into the area of fixed-parameter tractability and parameterized complexity (see, e.g., the excellent textbooks [DF99, FG06, Nie06] and the recent article by Buss and Islam [BI08]). For specific applications of fixed-parameter tractability and parameterized complexity to problems from computational social choice, we refer to the survey by Lindner and Rothe [LR08].

As another approach to coping with NP-hardness, we may sometimes be satisfied with having an efficient algorithm that doesn't yield the optimal solution but rather an approximation. While approximability has proven to be a natural and useful approach to coping with the NP-hardness of many important optimization problems in computer science (see, e.g., [ACG⁺03]), one may wonder whether it really is sensible in the context of voting. After all, election systems are used to determine an election winner, someone society finds most acceptable among all alternatives, and if an approximation algorithm instead of a real winner outputs some candidate whose score is no

worse than, say, one third of a real winner’s score (i.e., the algorithm approximates the optimal solution within a factor of three), then this candidate certainly should not be considered to be a reasonable social choice.

Nonetheless, approximation can be useful in the context of voting. For example, Brelsford et al. [BFH⁺08] set up a uniform framework that uses the approximability of NP-hard manipulation, control, and bribery problems as a measure of the effectiveness of specific manipulation, control, and bribery attacks. As another example, Caragiannis et al. [CCF⁺09] study the approximability of the winner problem for Dodgson and Young elections, two problems known to be NP-hard [BTT89b, HHR97, RSV03], and they propose two schemes for approximating Dodgson scores. They argue that such an “approximation algorithm is equivalent to a new voting rule” [CCF⁺09], and they prove their approximation schemes to possess certain properties that are desirable from a social-choice point of view.¹⁴ In contrast, they prove that Young scores are NP-hard to approximate within *any* factor.

LeGrand, Markakis, and Mehta [LMM06, LMM07] observed that there is an approximation algorithm for the minimax approval voting problem (which is defined below as the search problem corresponding to the decision problem FSM-AV), and we will present their algorithm here because it is related to the local search heuristics for this problem to be studied in Section 5.3.

First, we need some notation. The *weight* of a binary vector v , denoted by $weight(v)$, is the number of ones in v , and a k -*completion* of v is a vector v' that is obtained from v by randomly flipping $weight(v) - k$ ones to zeros if $weight(v) > k$, and by randomly flipping $k - weight(v)$ zeros to ones otherwise. Every k -completion v' of v has exactly k ones. Finally, given an election (C, V) and a vector $v \in \{0, 1\}^{|C|}$, we denote by $MaxHD_{(C, V)}(v) = \max_{b \in V} H(v, b)$ the maximum Hamming distance of v to the ballots in V .

The minimax approval voting problem seeks to find a minimax committee of weight k so as to minimize the maximum Hamming distance to all ballots in a given approval election. Formally, it is defined as follows (note that, since this is a search and not a decision problem, we have an “Output” field instead of a “Question” field):

Name: MINIMAX-AV.

Given: A set C of candidates, a collection V of voters represented via ballots over C , and a non-

¹⁴In particular, both approximation schemes proposed by Caragiannis [CCF⁺09] satisfy the Condorcet property [Con92] (see also [Fis77]), which says that a Condorcet winner—a candidate that defeats every other candidate by strict majorities—should be elected whenever one exists. Also, like many other natural voting rules, one of their approximation schemes is “weakly monotonic.”

negative integer $k \leq \|C\|$.

Output: A vector $v \in \{0, 1\}^{\|C\|}$ such that $\text{weight}(v) = k$ and $\text{MaxHD}_{(C,V)}(v)$ is minimum among the vectors in $\{0, 1\}^{\|C\|}$ of weight k .

Theorem 5.2 ([LMM06,LMM07]) *There is a polynomial-time algorithm that approximates the optimal solution of MINIMAX-AV within a factor of three.*

Proof. The approximation algorithm for MINIMAX-AV is rather simple. Given an election (C, V) and a nonnegative integer $k \leq \|C\|$, it works as follows:

1. Choose a ballot $v \in V$ at random. (Or, to keep this algorithm in deterministic polynomial time, select the first ballot.)
2. Compute a k -completion v' of v . (Or, to keep this algorithm in deterministic polynomial time, deterministically ensure in any simple way that v' via the smallest possible number of flips ends up having exactly k ones.)
3. Output v' as a solution.

Obviously, this algorithm runs in polynomial time. To estimate the approximation ratio of this algorithm, let $v \in V$ be the initial ballot chosen by the algorithm, let v' be the k -completion of v that is output by the algorithm, and let w be an optimal solution of MINIMAX-AV for the given input, i.e., $w \in \{0, 1\}^{\|C\|}$ is a vector of weight k such that $\text{MaxHD}_{(C,V)}(w)$ is minimum among all such vectors.

We have to show that for each $u \in V$, $H(v', u) \leq 3 \cdot \text{MaxHD}_{(C,V)}(w)$. Since the Hamming distance satisfies the triangle inequality, we have that for each $u \in V$,

$$H(v', u) \leq H(v', v) + H(v, u).$$

Applying the triangle inequality once again gives:

$$H(v', u) \leq H(v', v) + H(v, w) + H(w, u).$$

Since w is an optimal solution, it holds that $H(v, w) \leq \text{MaxHD}_{(C,V)}(w)$ and for all $u \in V$, $H(w, u) \leq \text{MaxHD}_{(C,V)}(w)$. Furthermore, since v' is a k -completion of v , $H(v', v) \leq H(w, v) \leq \text{MaxHD}_{(C,V)}(w)$. Combining these inequalities, we obtain the desired property:

$$H(v', u) \leq 3 \cdot \text{MaxHD}_{(C,V)}(w),$$

which proves that the approximation ratio of the above algorithm is three. □

5.3 Local Search Heuristics

The algorithm of Theorem 5.2 has a guaranteed approximation ratio of three even in the worst case, i.e., regardless of how unluckily we choose the initial vector and how unluckily we choose its k -completion. However, since the solution output by the algorithm depends solely on these random choices, one may wonder whether we perhaps can obtain a better result (i.e., a solution closer to the optimal solution) by picking the initial vector v and the bits to flip for computing a k -completion v' of v in a more purposeful or more clever way. In this section, we present some local search heuristics of LeGrand, Markakis, and Mehta [LMM06,LMM07] that are motivated by improving upon the approximation algorithm of Theorem 5.2. Although they don't have a better guaranteed approximation ratio in the worst case, these local search heuristics seem to perform better in practice.

In computer science, local search heuristics are often applied to find a good, even though typically not optimal, solution to a problem that is hard to compute. The starting point of a local search algorithm is some solution to the problem. Then the algorithm searches in an appropriate neighborhood of the starting solution to find a modification of this solution that is (in some sense) better. The general framework of a local search heuristic for the minimax problem can be described as follows: Given a set C of candidates, a set V of voters represented via ballots over C , a nonnegative integer $k \leq \|C\|$, and a parameter r , do the following:

1. Choose a vector v with $\text{weight}(v) = k$.
2. Repeat until $\text{MaxHD}_{(C,V)}(v)$ does not change anymore for at most $m = \|C\|$ iterations:
 - (a) Compute the set S of vectors resulting from flipping up to r bits in v from 0 to 1, and the same number of bits from 1 to 0.
 - (b) Compute the set $S' \subseteq S$ of vectors that minimize $\text{MaxHD}_{(C,V)}(s)$ for all vectors $s \in S$.
 - (c) Randomly choose, under the uniform distribution, some element of S' to be the new v .
3. Output v as a solution.

To show that such a heuristic algorithm runs in polynomial time on a machine that can make random choices, let us have a closer look at the second step first because we will present different approaches, with different running times, for the first step later on. In the second step, v is an m -bit vector, so $\text{MaxHD}_{(C,V)}(v)$ is between 0 and m , and for each of these values the loop in the second step may be executed m times in the worst case. Since $\text{MaxHD}_{(C,V)}(v)$ can only decrease and will never increase, the second step of the heuristic may be executed m times for each value, which

means the number of loop iterations is $\mathcal{O}(m^2)$. The set S can be computed in time $\mathcal{O}(m^{2r})$, since the number r of flipped bits is constant. To compute S' , $\text{MaxHD}_{(C,v)}(s)$ must be computed for all $s \in S$, which can be done in time $\mathcal{O}(mn)$. Altogether, the running time of the algorithm's second step is $\mathcal{O}(nm^{2r+3})$.

The adjustable parameters in this local search heuristic are the parameter r and the starting vector v . LeGrand, Markakis, and Mehta ([LMM06,LMM07]) proposed to choose $r = 1$ or $r = 2$ and proposed the following alternatives for the starting vector v :

1. A minisum solution, i.e., k candidates with the most approvals.
2. Choose a ballot v whose $\text{weight}(v)$ is closest to k so as to minimize the sum of the Hamming distances to all other ballots, and then compute a k -completion v' of v , where the bits to be flipped are chosen so as to minimize the sum of the Hamming distances to all other ballots.
3. A randomly chosen vector v with $\text{weight}(v) = k$.

Note that all these starting vectors can be computed in polynomial time (in models allowing appropriate access to randomness), so the heuristic runs in polynomial time (with access to random choices) for each of these choices. In Section 5.2, we saw that a minisum solution can be computed in deterministic polynomial time. Regarding the second choice of a starting vector, the vector v in the repeat-loop can be determined in polynomial time, since the weight of each ballot and the sum of the Hamming distances to all other ballots can be computed in polynomial time. The k -completion v' can also be computed in polynomial time, since it is the same as a computing a minisum solution, either with some fixed committee members (if $\text{weight}(v) < k$) or over a reduced candidate set (if $\text{weight}(v) > k$). Finally, note that the random choice of a vector v with $\text{weight}(v)$ for the third choice of a starting vector can obviously be done in polynomial time (with access to random choices).

We will refer to these heuristics by $h_{i,j}$, where $i \in \{1, 2, 3\}$ is the choice for the starting method named via the 3-item list's numbering above (so for example $i = 3$ means start with a randomly chosen vector v with $\text{weight}(v) = k$) and where j is the value chosen for the parameter r . The experimental evaluation with the data in Table 12 shows that these heuristics perform considerably better than the worst-case ratio of the approximation algorithm in Theorem 5.2 would suggest and also are significantly better than the minisum solution.

Let us explain these empirical results of LeGrand, Markakis, and Mehta [LMM06,LMM07] in a bit more detail. Table 12 considers three different types of elections, named “unbiased,” “biased,” and “GTS 2003,” where in each election the number of candidates is m , the committee size is k , and the number of ballots is n . The unbiased and biased elections consist of randomly generated

Election with m candidates, n ballots, and committee size k	unbiased		biased		GTS 2003
	$m = 20$	$m = 20$	$m = 20$	$m = 20$	$m = 24$
	$n = 50$	$n = 200$	$n = 50$	$n = 200$	$n = 161$
	$k = 10$	$k = 10$	$k = 10$	$k = 10$	$k = 12$
minimax	1.0000	1.0000	1.0000	1.0000	1.0000
minisum	1.1650	1.1521	1.2119	1.1932	1.2143
worst case	1.6746	1.4859	1.8509	1.6302	1.7143
$h_{1,1}$	1.0058	1.0320	1.0083	1.0210	1.0012
$h_{2,1}$	1.0118	1.0365	1.0112	1.0251	1.0017
$h_{3,1}$	1.0122	1.0370	1.0122	1.0262	1.0057
$h_{1,2}$	1.0004	1.0129	1.0004	1.0025	1.0000
$h_{2,2}$	1.0004	1.0164	1.0005	1.0029	1.0000
$h_{3,2}$	1.0004	1.0164	1.0005	1.0031	1.0000

Table 12: Average approximation ratios for local search heuristics obtained by LeGrand, Markakis, and Mehta [LMM06,LMM07].

ballots, each for 20 candidates and for a committee size of 10, and with the number n of ballots used being 50 or 200. In the unbiased case, the ballots are generated under the uniform distribution. In the biased case, the generation of the ballots is more complex: First, for each i , $1 \leq i \leq m$, two probabilities π_i and π'_i are randomly chosen for candidate c_i ; then, candidate c_i is approved of by 40% of the ballots with probability π_i and by another 40% of the ballots with probability π'_i , and the remaining 20% of the ballots are again chosen according to the uniform distribution. The GTS 2003 election, in contrast, is a real-life election performed by the Game Theory Society, with 24 candidates, a committee size of 12, and 161 ballots.

The entries in the “worst case” row of Table 12 represent the average approximation ratio of the worst outcome possible for the given election, i.e., a vector v with $weight(v) = k$ and such that $MaxHD_{(C,V)}$ is maximal. The entries in the minimax row are always 1.0000, since the minimax outcome is the optimal solution. All other entries give the average approximation ratios, always obtained from 5,000 simulations, where the four biased/unbiased elections are randomly chosen anew for each simulation but the GTS 2003 election is the same fixed election in each simulation.

To conclude, observe the following. First, Table 12’s empirical results are good for all starting vectors (i.e., the corresponding heuristics seem to perform significantly better than what one would expect from the worst-case approximation ratio of Theorem 5.2). Overall in Table 12, the $i = 1$

cases—i.e., using a minisum solution as the starting vector—seem to be the best choice among these heuristics. Second, the higher the r the better the solution, but the running time then also increases. Trivially, a sufficiently large r always leads to an optimal solution, since we can choose r so that the neighborhood inspected becomes the whole solution space (in that case, however, the heuristic would require the same amount of time as the naive algorithm needs for computing the optimal solution, which would render the heuristic useless).

Acknowledgments We are grateful to Claudia Lindner for many helpful proofreading comments and suggestions.

References

- [ACG⁺03] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, M. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation*. Springer-Verlag, second edition, 2003.
- [BF78] S. Brams and P. Fishburn. Approval voting. *American Political Science Review*, 72(3):831–847, 1978.
- [BF81] S. Brams and P. Fishburn. Approval voting, Condorcet’s principle, and runoff elections. *Public Choice*, 36(1):89–114, 1981.
- [BF83] S. Brams and P. Fishburn. *Approval Voting*. Birkhäuser, Boston, 1983.
- [BF02] S. Brams and P. Fishburn. Voting procedures. In K. Arrow, A. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, pages 173–236. North-Holland, 2002.
- [BF05] S. Brams and P. Fishburn. Going from theory to praxis: the mixed success of approval voting. *Social Choice and Welfare*, 25(2–3):457–474, 2005.
- [BFH⁺08] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, and I. Schnoor. Approximability of manipulating elections. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 44–49. AAAI Press, July 2008.
- [BI08] J. Buss and T. Islam. The complexity of fixed-parameter problems. *SIGACT News*, 39(1):34–46, 2008.
- [BKS04] S. Brams, D. Kilgour, and R. Sanver. A minimax procedure for negotiating multilateral treaties. In M. Wiberg, editor, *Reasoned Choices: Essays in Honor of Academy Professor Hannu Nurmi*, pages 108–139. The Finnish Political Science Association, 2004.

- [BKS07a] S. Brams, D. Kilgour, and R. Sanver. A minimax procedure for electing committees. *Public Choice*, 132(3–4):401–420, 2007.
- [BKS07b] S. Brams, D. Kilgour, and R. Sanver. A minimax procedure for negotiating multilateral treaties. In R. Avenhaus and I. Zartman, editors, *Diplomacy Games*, pages 265–282. Springer, 2007.
- [Bla58] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.
- [BO91] J. Bartholdi III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.
- [Bra80] S. Brams. Approval voting in multicandidate elections. *Policy Studies Journal*, 9(1):102–108, 1980.
- [BS06] S. Brams and R. Sanver. Critical strategies under approval voting: Who gets ruled in and ruled out. *Electoral Studies*, 25(2):287–305, 2006.
- [BS09] S. Brams and R. Sanver. Voting systems that combine approval and preference. In S. Brams, W. Gehrlein, and F. Roberts, editors, *The Mathematics of Preference, Choice, and Order: Essays in Honor of Peter C. Fishburn*, pages 215–237. Springer, 2009.
- [BTT89a] J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.
- [BTT89b] J. Bartholdi III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.
- [BTT92] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical Comput. Modelling*, 16(8/9):27–40, 1992.
- [CCF⁺09] I. Caragiannis, J. Covey, M. Feldman, C. Homan, C. Kaklamanis, N. Karanikolas, A. Procaccia, and J. Rosenschein. On the approximability of Dodgson and Young elections. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, January 2009. To appear.
- [Con92] J.-A.-N. de Caritat, Marquis de Condorcet. A survey of the principles underlying the draft constitution. 1792. Reprint by I. McLean and A. Urken, *Classics of Social Choice*, University of Michigan Press, 1995, pages 145–150.
- [CS03] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 781–788. Morgan Kaufmann, August 2003.

- [CS06] V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 627–634. AAAI Press, July 2006.
- [CSL07] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.
- [DF99] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, Berlin, Heidelberg, New York, 1999.
- [DKNS01] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference*, pages 613–622. ACM Press, 2001.
- [Dod76] C. Dodgson. A method of taking votes on more than two issues. Pamphlet printed by the Clarendon Press, Oxford, and headed “not yet published” (see the discussions in [MU95, Bla58], both of which reprint this paper), 1876.
- [DS00] J. Duggan and T. Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard–Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.
- [DSL06] B. Dutta, F. De Sinopoli, and J. Laslier. Approval voting: three examples. *International Journal of Game Theory*, 35:27–38, 2006.
- [EG08] U. Endriss and P. Goldberg, editors. *Proceedings of the 2nd International Workshop on Computational Social Choice*. University of Liverpool, 2008. Available online at www.csc.liv.ac.uk/~pwg/COMSOC-2008/proceedings.html.
- [EHRS07] G. Erdélyi, L. Hemaspaandra, J. Rothe, and H. Spakowski. On approximating optimal weighted lobbying, and frequency of correctness versus average-case polynomial time. In *Proceedings of the 16th International Symposium on Fundamentals of Computation Theory*, pages 300–311. Springer-Verlag *Lecture Notes in Computer Science* #4639, August 2007.
- [EHRS08] G. Erdélyi, L. Hemaspaandra, J. Rothe, and H. Spakowski. Frequency of correctness versus average-case polynomial time and generalized juntas. Technical Report TR-934, Department of Computer Science, University of Rochester, Rochester, NY, June 2008.
- [EKM07] P. Everaere, S. Konieczny, and P. Marquis. The strategy-proofness landscape of merging. *Journal of Artificial Intelligence Research*, 28:49–105, 2007.
- [EL05] E. Elkind and H. Lipmaa. Small coalitions cannot manipulate voting. In *Proceedings of the 9th International Conference on Financial Cryptography and Data Security*, pages 285–297. Springer-Verlag *Lecture Notes in Computer Science* #3570, 2005.

- [EL06] U. Endriss and J. Lang, editors. *Proceedings of the 1st International Workshop on Computational Social Choice*. Universiteit van Amsterdam, 2006. Available online at staff.science.uva.nl/~ulle/COMSOC-2006/proceedings.html.
- [ENR08a] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting broadly resists control. In *Proceedings of the 33rd International Symposium on Mathematical Foundations of Computer Science*, pages 311–322. Springer-Verlag *Lecture Notes in Computer Science* #5162, August 2008.
- [ENR08b] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. Technical Report arXiv:0806.0535v4 [cs.GT], ACM Computing Research Repository (CoRR), June 2008. Revised, September 2008.
- [ER97] E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.
- [Fal08] P. Faliszewski. Nonuniform bribery. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1569–1572. ACM Press, May 2008.
- [FG06] J. Flum and M. Grohe. *Parameterized Complexity Theory*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, 2006.
- [FHH06a] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 641–646. AAAI Press, July 2006.
- [FHH06b] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? Technical Report arXiv:cs/0608081v3 [cs.GT], ACM Computing Research Repository (CoRR), April 2006. Revised, August 2008.
- [FHHR07] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting broadly resist bribery and control. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 724–730. AAAI Press, July 2007.
- [FHHR08a] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Copeland voting fully resists constructive control. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, pages 165–176. Springer-Verlag *Lecture Notes in Computer Science* #5034, June 2008.
- [FHHR08b] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and control. Technical Report arXiv:0809.4484v2 [cs.GT], ACM Computing Research Repository (CoRR), September 2008.

- [FHHR09] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*. Springer, 2009. To appear. Available as Technical Report arXiv:cs/0609112v1 [cs.GT], ACM Computing Research Repository (CoRR), September 2006.
- [FHS08] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 983–990. ACM Press, May 2008.
- [Fis77] P. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.
- [Gib73] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.
- [GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.
- [GMHS99] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: The anatomy of recommender systems. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 434–435. ACM Press, 1999.
- [HH] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. *Journal of Heuristics*. To appear. Full version available as [HH05].
- [HH05] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. Technical Report TR-881, Department of Computer Science, University of Rochester, Rochester, NY, September 2005. Revised, June 2007.
- [HH06] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science*, pages 528–539. Springer-Verlag *Lecture Notes in Computer Science* #4162, August/September 2006.
- [HH07] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.
- [HHR97] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll’s 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.
- [HHR06] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. Technical Report

- arXiv:cs/0608057v2 [cs.GT], ACM Computing Research Repository (CoRR), August 2006.
- [HHR07a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.
 - [HHR07b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1308–1314. AAAI Press, January 2007.
 - [HO02] L. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, 2002.
 - [HSV05] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.
 - [KBS06] D. Kilgour, S. Brams, and R. Sanver. How to elect a representative committee using approval balloting. In F. Pukelsheim and B. Simeone, editors, *Mathematics and Democracy: Voting Systems and Collective Choice*, pages 83–95. Springer, 2006.
 - [LeG04] R. LeGrand. Analysis of the minimax procedure. Technical Report WUCSE-2004-67, Department of Computer Science and Engineering, Washington University, St. Louis, Missouri, November 2004.
 - [LMM06] R. LeGrand, E. Markakis, and A. Mehta. Approval voting: Local search heuristics and approximation algorithms. In U. Endriss and J. Lang, editors, *Proceedings of the 1st International Workshop on Computational Social Choice*, pages 276–289. Universiteit van Amsterdam, December 2006.
 - [LMM07] R. LeGrand, E. Markakis, and A. Mehta. Some results on approximating the minimax solution in approval voting. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1193–1195. ACM Press, May 2007.
 - [LR08] C. Lindner and J. Rothe. Fixed-parameter tractability and parameterized complexity, applied to problems from computational social choice. In A. Holder, editor, *Mathematical Programming Glossary*. INFORMS Computing Society, October 2008.
 - [MPRZ08] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. The complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, 2008.
 - [MPS08] J. McCabe-Dansted, G. Pritchard, and A. Slinko. Approximability of Dodgson’s rule. *Social Choice and Welfare*, 31(2):311–330, 2008.

- [MU95] I. McLean and A. Urken. *Classics of Social Choice*. University of Michigan Press, Ann Arbor, Michigan, 1995.
- [Nie06] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
- [Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [PR07] A. Procaccia and J. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, 28:157–181, 2007.
- [Rot05] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, 2005.
- [RSV03] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, 36(4):375–386, 2003.
- [Sat75] M. Satterthwaite. Strategy-proofness and Arrow’s conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.
- [ZPR08] M. Zuckerman, A. Procaccia, and J. Rosenschein. Algorithms for the coalitional manipulation problem. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms*, pages 277–286. Society for Industrial and Applied Mathematics, January 2008. Extended version to appear in *Artificial Intelligence*.