Separating and Collapsing Electoral Control Types

Benjamin Carleton
Department of Computer Science
Cornell University
Ithaca, USA
carleton@cs.cornell.edu

Michael C. Chavrimootoo
Department of Computer Science
University of Rochester
Rochester, USA
michael.chavrimootoo@rochester.edu

Lane A. Hemaspaandra
Department of Computer Science
University of Rochester
Rochester, USA
www.cs.rochester.edu/u/lane

David E. Narváez
Dept. of Electrical and Computer Eng.
Virginia Tech
Blacksburg, USA
david.narvaez@computer.org

Conor Taliancich
Property Matrix
Culver City, USA
ctalianc@u.rochester.edu

Henry B. Welles
Department of Computer Science
University of Rochester
Rochester, USA
hwelles@u.rochester.edu

ABSTRACT

Electoral control refers to attacking elections by adding, deleting, or partitioning voters or candidates [3]. Hemaspaandra et al. [16] discovered, for seven pairs $(\mathcal{T}, \mathcal{T}')$ of seemingly distinct standard electoral control types, that $\mathcal{T}$ and $\mathcal{T}'$ are identical: For each input I and each election system $\mathcal{E}$, I is a "yes" instance of both $\mathcal{T}$ and $\mathcal{T}'$ under $\mathcal{E}$, or of neither. Surprisingly, this had gone undetected even as the field was score-carding how many standard control types election systems were resistant to; various "different" cells on such score cards were, unknowingly, duplicate effort on the same issue. This naturally raises the worry that other pairs of control types are also identical, and so work still is being needlessly duplicated.

We determine, for all standard control types, which pairs are, for elections whose votes are linear orderings of the candidates, always identical. We show that no identical control pairs exist beyond the known seven. For three central election systems, we determine which control pairs are identical ("collapse") with respect to those particular systems, and we explore containment/incomparability relationships between control pairs. For approval voting, which has a different "type" for its votes, Hemaspaandra et al.'s [16] seven collapses still hold. But we find 14 additional collapses that hold for approval voting but not for some election systems whose votes are linear orderings. We find one additional collapse for veto and none for plurality. We prove that each of the three election systems mentioned have no collapses other than those inherited from Hemaspaandra et al. [16] or added here. But we show many new containment relationships that hold between some separating control pairs, and for each separating pair of standard control types classify its separation in terms of containment (always, and strict on some inputs) or incomparability.

Our work, for the general case and these three important election systems, clarifies the landscape of the 44 standard control types, for each pair collapsing or separating them, and also providing finer-grained information on the separations.

KEYWORDS

Computational Social Choice; Elections and Voting; Electoral Control

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

ACM Reference Format:
Benjamin Carleton, Michael C. Chavrimootoo, Lane A. Hemaspaandra, David E. Narváez, Conor Taliancich, and Henry B. Welles. 2023. Separating and Collapsing Electoral Control Types. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Suppose Professor Fou specializes in complexity classification, and for each problem that comes through the door tries to prove it complete or not for Fou's pet list of classes: NP, coNP, and cocoNP. So for years problems from computational social choice or elsewhere might come in the door. And Fou might separately prove them complete for NP and cocoNP, and might give evidence suggesting coNP-completeness is wildly unlikely. Most CS academics would be horrified at the situation and would say: How can Fou not have stepped back and tried to see the big picture—and realized that NP and cocoNP are the same class, and thus that Fou was doing duplicate work! This seems at first a comic situation, yet a more subtle cousin of it has been in play in the computational social choice world for many years.

Let us explain what we mean by that. Control and manipulation are the two families of attack types that were the focus of the seminal papers of Bartholdi et al. [1, 2, 3]. The various control attacks model different attempts to affect the outcome of elections by changes to their structure, namely, via adding, deleting, or partitioning voters or candidates. The control attack types also vary as to whether the goal is to make the focus candidate a winner (perhaps tied; this is known as the nonunique-winner (NUW) or cowinner setting), or to be a winner who is not tied with anyone else (this is known as the unique-winner (UW) setting), or to *not* (again regarding one of those two variants regarding ties) win. Over time, there has been something of a race or contest in computational social choice to find election systems for which a very large number of control types have the property that it is NP-hard to determine on a given input whether that type of control can succeed (see the discussion in our Related Work section).

But what if two (compatible, i.e., having the same input type) control types were in fact the same? That is, what if for every election system (whose votes are linear orders over the candidates) and on every input, despite the control types seeming to model different actions, in fact either for both control types the answer is yes (i.e., for each of the two control types, there is a way to achieve the goal of making the focus candidate win (or lose)), or for both the answer is no (i.e., for each of the two control types, there is no way to achieve the goal of making the focused candidate win (or lose)). We will say that the types "collapse" in this case. Then the two types *for all practical purposes* are the same. In such a case, all research that separately studied the two types—for example to determine their computational complexity for some election system (over linear orders)—would be doing the same work twice. As our abstract mentioned, surprisingly, it has recently been observed by Hemaspaandra et al. [16] that there are (at least) seven such collapsing pairs of control types. In fact, that paper shows that four control types pairwise collapse—yielding six collapsing pairs—and one other pair also collapses.

The natural question this raises is whether there are other undiscovered collapses—either ones that hold for all election systems or, failing that, ones that hold when our focus is limited to one of the most important election systems. We completely resolve that question, both as to whether any additional collapses beyond the seven hold in general—we show that the remaining hundreds of pairs all separate—and as to the major election systems plurality, veto, and approval—for two of which we do discover additional collapses (one additional collapsing pair for veto and 14 additional collapsing pairs for approval) and for all three of which we then separate all remaining pairs. The universe of control types we study is that of the 22 "standard" control types used by Hemaspaandra et al. [16] and other papers, each studied in both the NUW and the UW winner model; so 44 control types in total.

Why is this important? Collapses reveal that seemingly different control types are the same ones in disguise. So general collapses give a clearer picture of the world of control types, and can help us avoid duplicate work.[1] Separations on the other hand assure us that the types differ.

Since (general-case) collapse is defined as universal over election systems, its negation (separation) is existential over election systems, and as this paper itself shows, specific systems may have additional collapses. Thus it is important for the field, as to its understanding of control types, to find the collapse/separation behavior for important election systems, and we completely resolve this for three central ones: plurality, veto, and approval. We hope that future papers by others will study additional systems or, even better, will (as we do in Section 3.3) define sufficient conditions that in one fell swoop provide collapses for groups of election systems.

Separations themselves can be refined, for two (compatible) control types, if on each input (except with no focus candidate specified in the input) we consider for each of the two control types the set of all candidates c such that if c is the focal candidate, the given control action can succeed. It might be the case that for the first control type on all inputs that set is a subset of that set for the second control type; or it could always be a superset; or neither of these cases could hold. Of course, if the first two cases hold simultaneously, that is the same as the two types collapsing. For *all* our control pairs that separate, we refine our separations into the three cases just described: "always $\subseteq$, and on some instance $\subsetneq$"; "always $\supseteq$, and on some instance $\supsetneq$"; and neither of the above (i.e., "incomparable"). We find, for various separating pairs, such containment relationships, and not all are UW and NUW variants of each other (though UW and NUW always have an obvious $\subseteq$ or $\supseteq$ relation; which one of those holds depends on whether the type is "constructive" ($\subseteq$ holds) or "destructive" ($\supseteq$ holds); we will refer back to this fact later as $(**)$).

As to proof techniques, for separations, since the number of pairs is huge, when we can we build examples that simultaneously yield many separations. Some of our separation constructions are obtained by computer-aided search. (For information on reproducibility, verifications, and availability of code, see Additional Note 1, which as described in the next paragraph is available in our public repository.) Some of our separation results on the "general-case" tables are obtained by inheritance from a specific-case table. As to new collapses, some are achieved by direct arguments that exploit some feature of the setting, and most are proven via an axiomatic-sufficient-condition approach. Also, some of the collapse (equality) entries in our tables for specific election systems are inherited from the general-case tables.

Big picture, this paper completely resolves the extent to which control-type pairs collapse or fail to collapse, both in general (i.e., universally quantified over election systems) and with regard to plurality, veto, and approval elections. *We include four proofs in the main body. All regard approval, as approval has the most collapses among the three important election systems we studied, and the proofs regarding approval are quite varied. The other proofs can be found in our technical report version [6]. Table 1 is included in this paper, and all other tables referenced or mentioned in this paper—even ones mentioned as "our tables" or "the tables"—can be found, with the same table numberings as referenced here, in that technical report. Also, to support reproducibility, and to aid researchers who might wish to carry our study to other cases, we have made all our computer-search programs, and their inputs and outputs, publicly available in an online repository (https://github.com/MikeChav/SCT_Code). Each "Additional Note" referenced in this paper can be found in that repository's pdf file of Additional Notes for this paper.*

2 PRELIMINARIES

For consistency, some of the standard definitions that appear in this section are taken, at times verbatim, from [16]. An election comprises a finite set C of candidates (each identified uniquely by a name[2]) and a finite collection V of votes over C. Except in one section of the paper (where we will study a system using a different vote type, known as approval vectors), we throughout this paper take the "type" of a vote as being a linear ordering over

[1]Such duplicate work has already occurred extensively. Each time a paper built polynomial-time algorithms for both elements of a collapsing pair of control types, or proved NP-completeness for both elements, the paper did needless work on one element of the pair, since the sets involved in such a pair are the same set and so perforce they are of identical complexity. As just a few of the many papers that would have been saved a bit or a lot of work by either the seven general collapses of [16] or the additional concrete-system collapses established in the present paper, we mention [13, 17, 21, 23].

[2]By allowing names we potentially allow the names to be nefariously exploited by election systems. However, that model as to the use of names in fact makes our collapses and containments *stronger* than if those results were in a model where (candidate-)neutrality is assumed/required (the same applies to all collapses from Hemaspaandra et al. [16], as those results are in this same with-names model). Although use of names would make our separations weaker, we address that by having ensured that every separation we establish in this paper is achieved via a (candidate-)neutral election system. In contrast, as mentioned in our Related Work section, the one separation proven in Hemaspaandra et al. [16] uses a system that is not (candidate-)neutral.

C (note: linear orderings—complete, transitive, antisymmetric binary relations—are inherently tie-free). A simple example of a 3-candidate, 4-voter election thus is $C = \{a, b, c\}$ and the votes being $a > b > c$, $a > b > c$, $a > c > b$, and $b > c > a$ (see Additional Note 2 for additional discussion of the model used for votes).

As is standard in computational social choice, *an election system $\mathcal{E}$ maps an election (a pair (C, V)) to a (possibly nonstrict) subset of C (the set of winners)*. As is often done in computational social choice papers, we do not forbid the case of having no winners; see Hemaspaandra et al. [16, Footnote 3] for a discussion of why allowing that is natural.

Studying electoral control from a computational perspective was initiated by Bartholdi et al. [3]. Their notion, known as constructive control, focuses on making a particular candidate a winner by some "control" action. Hemaspaandra et al. [17] define the natural variants of those where instead the goal is to *prevent* a particular candidate from winning. This is known as destructive control.

The control actions are: partition of voters, partition of candidates, run-off partition of candidates, deleting voters, deleting candidates, adding voters, and adding candidates (both limited and unlimited). For each partition-based control type, we have two subvariants to handle the outcome of the subelections: in the TE (ties eliminate) subvariant, a candidate proceeds to the final round exactly if that candidate is the unique winner of that subelection, and in the TP (ties promote) subvariant, every winner of a subelection (including nonunique winners) proceeds to the final round. Hemaspaandra et al. [16] suggest it is more natural to study the TE subvariant when dealing with the unique-winner model and the TP subvariant when dealing with the nonunique-winner model. Since our goal is to uncover all possible collapses and separations (within the standard control types and their variants), we consider both TE and TP subvariants, regardless of the winner model.

Definition 1 (see Hemaspaandra et al. [16] and Additional Note 3). *Let $\mathcal{E}$ be an election system.*

(1) In the ***constructive control by adding candidates*** *problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-AC-NUW), we are given two disjoint sets of candidates C and A, V a collection of votes over $C \cup A$, a candidate $p \in C$, and a nonnegative integer k. We ask if there is a set $A' \subseteq A$ such that (i) $\|A'\| \leq k$ and (ii) p is a winner of $\mathcal{E}$ election $(C \cup A', V)$.*

(2) In the ***constructive control by unlimited adding candidates*** *problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-UAC-NUW), we are given two disjoint sets of candidates C and A, V a collection of votes over $C \cup A$, and a candidate $p \in C$. We ask if there is a set $A' \subseteq A$ such that p is a winner of $\mathcal{E}$ election $(C \cup A', V)$.*

(3) In the ***constructive control by deleting candidates*** *problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-DC-NUW), we are given an election (C, V), a candidate $p \in C$, and a nonnegative integer k. We ask if there is a set $C' \subseteq C$ such that (i) $\|C'\| \leq k$, (ii) $p \notin C'$, and (iii) p is a winner of $\mathcal{E}$ election $(C - C', V)$.*

(4) In the ***constructive control by adding voters*** *problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-AV-NUW), we are given a set of candidates C, two collections of votes, V and W, over C, a candidate $p \in C$, and a nonnegative integer k. We ask if there is a collection $W' \subseteq W$ such that (i) $\|W'\| \leq k$ and (ii) p is a winner of $\mathcal{E}$ election $(C, V \cup W')$.*

(5) In the ***constructive control by deleting voters*** *problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-DV-NUW), we are given an election (C, V), a candidate $p \in C$, and a nonnegative integer k. We ask if there is a collection $V' \subseteq V$ such that (i) $\|V'\| \leq k$ and (ii) p is a winner of $\mathcal{E}$ election $(C, V - V')$.*

(6) In the ***constructive control by partition of voters*** *problem for $\mathcal{E}$, in the TP or TE tie-handling rule model (denoted by $\mathcal{E}$-CC-PV-TP-NUW or $\mathcal{E}$-CC-PV-TE-NUW, respectively), we are given an election (C, V), and a candidate $p \in C$. We ask if there is a partition[3] of V into V_1 and V_2 such that p is a winner of the two-stage election where the winners of subelection (C, V_1) that survive the tie-handling rule compete (with respect to vote collection V) along with the winners of subelection (C, V_2) that survive the tie-handling rule. Each election (in both stages) is conducted using election system $\mathcal{E}$.*

(7) In the ***constructive control by run-off partition of candidates*** *problem for $\mathcal{E}$, in the TP or TE tie-handling rule model (denoted by $\mathcal{E}$-CC-RPC-TP-NUW or $\mathcal{E}$-CC-RPC-TE-NUW, respectively), we are given an election (C, V), and a candidate $p \in C$. We ask if there is a partition of C into C_1 and C_2 such that p is a winner of the two-stage election where the winners of subelection (C_1, V) that survive the tie-handling rule compete (with respect to vote collection V) against the winners of subelection (C_2, V) that survive the tie-handling rule. Each election (in both stages) is conducted using election system $\mathcal{E}$.*

(8) In the ***constructive control by partition of candidates problem*** *for $\mathcal{E}$, in the TP or TE tie-handling rule model (denoted by $\mathcal{E}$-CC-PC-TP-NUW or $\mathcal{E}$-CC-PC-TE-NUW, respectively), we are given an election (C, V), and a candidate $p \in C$. We ask if there is a partition of C into C_1 and C_2 such that p is a winner of the two-stage election where the winners of subelection (C_1, V) that survive the tie-handling rule compete (with respect to vote collection V) against all candidates in C_2. Each election (in both stages) is conducted using election system $\mathcal{E}$.*

There are 11 control "types" listed above (applied regarding generic election system $\mathcal{E}$). For each, we can change "is a winner" to "is an untied (i.e., unique) winner"; for those 11 variants, we change the "-NUW" into "-UW." Thus we have 22 control types. Those 22 are all trying to make the focus candidate win. And so they are all spoken of as "constructive" control types (thus the "CC" in their naming strings). Finally, for each of those now 22 control types, we can ask whether one can ensure that the focus candidate is *not* a winner or *not* a unique winner; those are known as the "destructive" control variants, and in the names of those, the CC is replaced by a DC, e.g., $\mathcal{E}$-DC-AC-UW.

We thus have 44 total types of control, which we will view as the "standard" control types. We thus have $\binom{44}{2} = 946$ pairs of control types. Fortunately, 624 of those pairs are incompatible—the two control types have different input fields from each other[4] and

[3]A partition of a collection V is a pair of collections V_1 and V_2 such that $V_1 \cup V_2 = V$, where $\cup$ denotes multiset union. A partition of a set C is a pair of sets C_1 and C_2 such that $C_1 \cup C_2 = C$ and $C_1 \cap C_2 = \emptyset$, where $\cup$ and $\cap$ are standard set union and intersection.

[4]So for example CC-PC-TP-NUW and CC-RPC-TE-UW are compatible, but CC-PC-TP-NUW and CC-AV-NUW are not, since CC-AV-NUW has a nonnegative integer

so comparing them would not even make sense—and so we will exclude them from our study. So we have "only" 322 pairs to focus on in our study. Table 2 shows the five compatibility equivalence classes that the 44 types partition into.

We often when speaking of control types will be speaking of the control model itself, and when doing so, we generally do not include the "$\mathcal{E}$-" prefix. In some sense, we view, e.g., CC-AC-NUW as a control type (model)—one among the 44 standard such models—and *we view* $\mathcal{E}$-CC-AC-NUW *as the set of input strings that are "yes" instances, for election system* $\mathcal{E}$*, under that model of control.*

Keeping that in mind, we now define precisely what we mean by two (compatible) control types collapsing or separating, and then introduce a function approach that will allow us to explore in a more refined way the nature of the separations. Let $\mathcal{E}$ be an election system (e.g., Plurality) and let $\mathcal{T}$ and $\mathcal{T}'$ be two (compatible) control types from our 44 standard ones (e.g., CC-AC-NUW and CC-AC-UW). Then if $\mathcal{E}$-$\mathcal{T}$ = $\mathcal{E}$-$\mathcal{T}'$ we say that the control types $\mathcal{T}$ and $\mathcal{T}'$ collapse (for election system $\mathcal{E}$), and if $\mathcal{E}$-$\mathcal{T} \neq \mathcal{E}$-$\mathcal{T}'$ we say that the control types $\mathcal{T}$ and $\mathcal{T}'$ separate (for election system $\mathcal{E}$). We will also use the terms collapse and separate for a more general case, namely, the one of quantifying universally over all election systems whose votes are linear orders. When speaking in that case, we will say that two (compatible) control types from among our 44 collapse if the two control types collapse for every election system $\mathcal{E}$ whose vote type is linear orders, and otherwise we will say that the control types separate.

As an example, if we fix the election system $\mathcal{E}$ to be plurality, we note that in the 3-candidate, 4-voter election example given at the beginning of this section a is a unique winner. Yet, we could make b a unique winner by deleting candidate a, hence we have $(C, V, b, 1) \in$ Plurality-CC-DC-UW. Also, since a unique winner is a winner in the nonunique-winner model as well, we have $(C, V, b, 1) \in$ Plurality-CC-DC-NUW. (In this paper we will not focus on encoding details, since they are not important to our study.)

Control types that separate can do so in different ways, some of which reflect containment relationships. In order to be able to seek such relationships, we define the three different ways that two (compatible) separating control types, $\mathcal{T}$ and $\mathcal{T}'$, for $\mathcal{E}$ can separate. To support this refinement, we introduce a function model. In particular, we will define functions that, for a given control type, map from inputs (that differ from those used so far for that control type only in *not* having a focus candidate specified) to the set of all candidates c such that if c is made the focus candidate for that input, successful control is possible. In a bit more detail: Each of our control types has certain inputs, and all include a focus candidate, c. Let us for any of our 44 control types, $\mathcal{T}$, refer to an input to it, except with the field containing the focus candidate removed, as the reduced form of that input; and if we wanted to add back the name of a particular candidate to be the focus candidate, we will say that is inflating the reduced input by that candidate. For any of our 44 control types, $\mathcal{T}$, and any election system $\mathcal{E}$, we define the function $f_{\mathcal{E}\text{-}\mathcal{T}}$ to be the function that, for a given reduced input $\tilde{I}$, outputs the set of all candidates c such that $\tilde{I}$ inflated by c belongs to the set $\mathcal{E}$-$\mathcal{T}$. For example, given as input $(C, V, 3)$, the output of $f_{\mathcal{E}\text{-DC-DV-NUW}}$ is the set of all candidates c that can by deleting less than or equal to three votes be prevented from being a winner.

Clearly, two (compatible) control types collapse exactly if their thus-defined functions are equal. But for two separating (compatible) control types, there are three different ways they can be separated. One way is if, for each reduced input $\tilde{I}$: (a) $f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) \subseteq f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I})$ and (b) for some reduced input $\tilde{I}'$, $f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}') - f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}') \neq \emptyset$. We will (in a slight abuse of notation) refer to the case where (a) holds as the "$\subseteq$" case, and the case where both (a) and (b) hold as the "$\subsetneq$" case. The "$\supseteq$" and "$\supsetneq$" cases are analogously defined. If the two (compatible) types separate but neither the "$\subsetneq$" case nor the "$\supsetneq$" case holds, we will say the two types are incomparable: each will sometimes have successful focus candidates that the other does not. If both directions of noncontainment can be witnessed by the same reduced input, we will say the two types are strongly incomparable, i.e., there is a reduced input $\tilde{I}$ such that $f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) - f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}) \neq \emptyset$ and $f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}) - f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) \neq \emptyset$. These notions are relative to each specific election system, $\mathcal{E}$.

For the general case—where our collapses are universally quantified over all election systems whose vote type is linear orders—we define three increasingly strong types of incomparability. The weakest incomparability notion is simply that for at least one such election system $\mathcal{E}$, for some reduced input $\tilde{I}$, $f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) - f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}) \neq \emptyset$ holds, and for at least one such election system $\mathcal{E}'$, for some reduced input $\tilde{I}'$, $f_{\mathcal{E}'\text{-}\mathcal{T}'}(\tilde{I}') - f_{\mathcal{E}'\text{-}\mathcal{T}}(\tilde{I}') \neq \emptyset$ holds. (Note: We will state no weak incomparability results in this paper, since whenever we obtained a weak incomparability result we in fact were able to even establish incomparability.) We will call this being weakly incomparable. We say the pair is incomparable if there is some election system, over linear orders, in which the pair is incomparable in the sense of the previous paragraph. And we will say the pair is strongly incomparable if there is some election system, over linear orders, in which the pair is strongly incomparable in the sense of the previous paragraph. For the general case, we will say "$\subseteq$" (resp., "$\supseteq$") holds if for every election system $\mathcal{E}$ whose vote type is linear orders, the "$\subseteq$" (resp., "$\supseteq$") case for $\mathcal{E}$, as defined above, holds.

We now cover, and give brief reference labels to (though we at times may use these inheritances tacitly when the use is clear), some collapse, containment, and separation inheritances that hold. General-case collapse, $\subseteq$, and $\supseteq$ results imply, for each election system over linear orders, resp., collapse, $\subseteq$, and $\supseteq$ results (let us shorthand that fact as I1). If a given general case result of this type *happens* to in addition hold for each election system (not merely those over linear orders), we will refer to that as additionally being an I1$^+$ case. Because their particular proofs do not ever draw on the vote types, the seven general-case collapses of Hemaspaandra et al. [16], and also our general containment results of Theorem 2, hold even as I1$^+$ cases. As to separation inheritances, if for some election system $\mathcal{E}$ over linear orders and some reduced input $\tilde{I}$ we have that $f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}) - f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) \neq \emptyset$ (as, crucially, is always the case if we have that for $\mathcal{E}$ the $\subsetneq$ relation holds between $\mathcal{T}$ and $\mathcal{T}'$) and we in addition happen to have that in the general case the $\subseteq$ relation is known to hold between $\mathcal{T}$ and $\mathcal{T}'$, then we may conclude that $\subsetneq$ holds in our general case; the analogous claim holds for the $\supsetneq$ case, and we will refer to either of these, when they hold, as an I2 inheritance case. Incomparability in an election system (over linear

field k. Throughout this paper, whenever we speak of pairs of control types, we refer only to compatible pairs (even if that is not explicitly stated, although for emphasis we often do state it).

orders) implies general-case incomparability (we will refer to this as I3), and strong incomparability in an election system (over linear orders) implies general-case strong incomparability (I3*).

3 RESULTS

In the following three subsections, for the general case and for the cases of plurality, veto, and approval voting, we completely determine which pairs of (compatible) standard control types collapse, and which separate. Beyond that, we refine every separation into one of the three disjoint cases: $\subsetneq$, $\supsetneq$, and incomparability.

We discover a number of previously unknown collapses (for two of the three specific systems) and also for many noncollapsing control-type pairs establish new containments in one direction. *Regarding the former, for veto we discover a new collapsing pair and for approval we extend a four-control type collapse by Hemaspaandra et al. [16] to a six-control type collapse, and we find five additional collapsing pairs, for a total of 14 new collapsing control-type pairs for approval. Overall, we establish that, for the* $4 \cdot 322 = 1{,}288$ *cases we study (all compatible pairs for each of the four cases), no containments or collapses exist other than those provided by either Hemaspaandra et al. [16] or this paper.*

3.1 The General Case and Plurality

We show that the only (compatible) pairs that collapse in general (i.e., that collapse for every election system) are the seven found by Hemaspaandra et al. [16], namely, for every election system $\mathcal{E}$, $\mathcal{E}$-DC-RPC-TE-NUW = $\mathcal{E}$-DC-RPC-TE-UW = $\mathcal{E}$-DC-PC-TE-NUW = $\mathcal{E}$-DC-PC-TE-UW (six pairs) and $\mathcal{E}$-DC-RPC-TP-NUW = $\mathcal{E}$-DC-PC-TP-NUW (one pair). Surprisingly, we will be able to do so using just constructions about plurality elections, combined with uses of inheritance.

In a plurality election, for each vote where a particular candidate is ranked first, that candidate receives a point, and a winner is a candidate with the highest number of points among all the candidates (naturally, there can be multiple winners). The votes in plurality elections are linear orders. Recall that by the "general case," we mean the general case with respect to elections where the votes are linear orders. Thus to separate two control types in the general case, it certainly suffices to show that they separate under plurality, as such separations inherit back to the general case via the I3 and I2 inheritance frameworks from our preliminaries (see also Additional Note 4). The reason we know that no separation for the general case is missed is that our results show that plurality has not a single $\subsetneq$, $\supsetneq$, or collapse result not also possessed by the general case; so our I2 cases are valid, and every general-case $\subsetneq$, $\supsetneq$, or incomparability (or even strong incomparability) that holds is yielded by our inheritances.

Let us now turn our attention to the two new general-case containments shown in this paper (these containments are in addition to the obvious ones, noted at location (∗∗) of the introduction, about two control types that only differ in their winner model) and also argue that both are strict in the general case (by which, recall, we mean that there is at least one election system under which the containment is not an equality). The following two containments apply to all vote types (not just linear orders; and we argue in Section 3.3 that the collapses by Hemaspaandra et al. [16] also apply to all vote types).

Theorem 2. *Let $\mathcal{E}$ be an election system. For each $\mathcal{T} \in$ {DC-RPC-TP-UW, DC-PC-TP-UW}, $\mathcal{E}$-$\mathcal{T} \subseteq \mathcal{E}$-DC-RPC-TE-NUW.*

For both containments above, Table 5 contains a pointer to a separation witness in Table 4 that shows that the containment is not an equality in the general case (i.e., that for some election system—in fact, plurality—the containment is strict). More generally, Table 5, for each compatible pair of control types $\mathcal{T}$ and $\mathcal{T}'$, gives us an election witnessing Plurality-$\mathcal{T}$ − Plurality-$\mathcal{T}' \neq \emptyset$ if that holds, and gives an election witnessing Plurality-$\mathcal{T}'$ − Plurality-$\mathcal{T} \neq \emptyset$ if that holds (and if both hold Table 5 gives witnesses for each).

Some of the constructions in Table 4 are quite simple. For example, with $C = \{a, b\}$ and $V = \{a > b\}$ (denoted by "Plur.3") we clearly get incomparability between all 144 pairs of partition types where one type is constructive and the other is destructive. (This holds since in that election, under every constructive or destructive partition-control action, a is the unique winner and b is not a winner.) However, showing separations for every pair that can be separated is no trivial matter. Some of our separation examples for plurality are quite large, with up to 18 votes and up to seven candidates. The search for those examples was computer-aided and extensive.

3.2 Veto

In a veto election, for each vote where a candidate is not ranked last, that candidate receives a point, and a winner is a candidate with the highest number of points among all the candidates (naturally, there can be multiple winners). For example, if $C = \{a, b, c\}$ and $V = \{a > b > c, c > a > b\}$, then b and c each receive one point, but a receives two and wins. *We now establish every equality and containment that holds for veto elections but was not established by one of the results of Hemaspaandra et al. [16]. For all other veto cases, we have constructed counterexamples.* Some of those counterexamples were obtained through computer search.

Theorem 3. Veto-DC-PV-TE-UW = Veto-DC-PV-TE-NUW.

Theorem 4. *For each $\mathcal{T} \in$ {DC-PV-TP-UW, DC-PV-TP-NUW},* Veto-$\mathcal{T} \subsetneq$ Veto-DC-PV-TE-NUW.

Theorem 5. *For each $\mathcal{T} \in$ {DC-RPC-TE-NUW, DC-RPC-TP-UW, DC-RPC-TP-NUW, DC-PC-TP-UW},* Veto-$\mathcal{T} \subsetneq$ Veto-DC-PV-TE-NUW.

3.3 Approval Voting

Approval voting differs from the other election systems discussed so far in this paper as to its vote type. In an approval election (C, V), each vote is a bit-vector of length $\|C\|$, with each bit being associated with a candidate. If a candidate's bit is 1, then that candidate is approved by that vote. In approval voting, the winner set is composed of each candidate c such that no other candidate is approved by strictly more votes than c is. We will sometimes speak of the "score" of a candidate in the rest of this section. In the context of approval, the score of a candidate in an election is the number of votes that approve that candidate (and as noted above, the set of winners are those candidates with maximal score).

Although the collapses shown by Hemaspaandra et al. [16] were stated for election systems where votes are linear orders, we note that in their proof they do not use the vote type and thus those collapses hold regardless of what type of votes the election system is over. Thus we have the following corollary.

Corollary 6 (see [16]). *(1)* Approval-DC-RPC-TE-UW = Approval-DC-RPC-TE-NUW = Approval-DC-PC-TE-NUW = Approval-DC-PC-TE-UW. *(2)* Approval-DC-RPC-TP-NUW = Approval-DC-PC-TP-NUW.

In this section, we prove a number of new collapses and inclusions that hold for approval voting. Some of these results draw on previously established immunity[5] arguments that draw on axioms satisfied by approval voting, thereby allowing us to generalize our results. Other results are provided by direct arguments. Our direct arguments often rely on the fact that, since the votes are bit-vectors, a candidate's score in an election is independent of the other candidates present in the election. We will start by discussing the results that draw on the aforementioned immunity arguments.

Let us first consider the Weak Axiom of Revealed Preferences (WARP), which requires that p winning an election (C, V) implies that p remains a winner of every election (C', V) for which $p \in C' \subseteq C$ [25]. The "unique" version of this axiom (Unique-WARP) only differs in that it requires p to be a unique winner, i.e., it requires that p uniquely winning in an election (C, V) implies that p uniquely wins in each election (C', V) for which $p \in C' \subseteq C$. [17] notes that any election system that satisfies Unique-WARP is immune to several types of control, namely to (i) destruc. control by partition of candidates and run-off partition of candidates (under both TP and TE tie-handling rules) in the UW model, and (ii) destruc. control by deleting candidates in the UW model. Using this, our proofs are able to pinpoint the exact content of the relevant sets and establish the following results.[6]

Theorem 7. *Let $\mathcal{E}$ be an election system satisfying Unique-WARP. Then* $\mathcal{E}$-DC-PC-TP-UW = $\mathcal{E}$-DC-PC-TE-UW (= $\{(C, V, p) \mid p \in C$ *and p is not a unique winner of $\mathcal{E}$ election* $(C, V)\}$).

Proof. Fix any $\mathcal{E}$ that satisfies Unique-WARP. Let $\mathcal{E}$-$\mathcal{T}_1$ = $\mathcal{E}$-DC-PC-TP-UW and let $\mathcal{E}$-$\mathcal{T}_2$ = $\mathcal{E}$-DC-PC-TE-UW. Consider the two sets $A_{\mathcal{E}} = \{(C, V, p) \mid p \in C$ and p is a unique winner of $\mathcal{E}$ election $(C, V)\}$ and $B_{\mathcal{E}} = \{(C, V, p) \mid p \in C$ and p is not a unique winner of $\mathcal{E}$ election $(C, V)\}$. These sets form a partition of $Y = \{(C, V, p) \mid p \in C\}$, so of course $Y = A_{\mathcal{E}} \cup B_{\mathcal{E}}$. Clearly we also have that $\mathcal{E}$-$\mathcal{T}_1 \subseteq Y$ and that $\mathcal{E}$-$\mathcal{T}_2 \subseteq Y$. We will argue that $\mathcal{E}$-$\mathcal{T}_1 = B_{\mathcal{E}} = \mathcal{E}$-$\mathcal{T}_2$. Since $\mathcal{E}$, like all systems satisfying Unique-WARP, is immune to both control types in the theorem statement, we have (recall that both of these types are destructive types) that $\mathcal{E}$-$\mathcal{T}_1 \cap A_{\mathcal{E}} = \emptyset$ and $\mathcal{E}$-$\mathcal{T}_2 \cap A_{\mathcal{E}} = \emptyset$, and thus it holds that $\mathcal{E}$-$\mathcal{T}_1 \subseteq B_{\mathcal{E}}$ and $\mathcal{E}$-$\mathcal{T}_2 \subseteq B_{\mathcal{E}}$. Fix $(C, V, p) \in B_{\mathcal{E}}$. Then the partition $(\emptyset, C)$ witnesses both $(C, V, p) \in \mathcal{E}$-$\mathcal{T}_1$ and $(C, V, p) \in \mathcal{E}$-$\mathcal{T}_2$, since in both cases the final round will simply be (C, V) and we know that since $(C, V, p) \in B_{\mathcal{E}}$, p will not be a unique winner of the final round. ❑

Theorem 8. *Let $\mathcal{E}$ be an election system that satisfies Unique-WARP. Then the following hold. (1)* $\mathcal{E}$-DC-DC-UW $\subseteq$ $\mathcal{E}$-DC-DV-UW. *(2)* $\mathcal{E}$-DC-DC-NUW $\subseteq$ $\mathcal{E}$-DC-DV-UW.

Theorem 9. *Let $\mathcal{E}$ be an election system that satisfies WARP. Then* $\mathcal{E}$-DC-DC-NUW $\subseteq$ $\mathcal{E}$-DC-DV-NUW.

Theorem 10. *Let $\mathcal{E}$ be an election system that satisfies Unique-WARP. Then* $\mathcal{E}$-CC-PC-TP-UW = $\mathcal{E}$-CC-RPC-TP-UW.

Proof. We use an argument similar to that of Thm. 7. Fix any $\mathcal{E}$ that satisfies Unique-WARP. Let $\mathcal{E}$-$\mathcal{T}_1$ = $\mathcal{E}$-CC-PC-TP-UW and let $\mathcal{E}$-$\mathcal{T}_2$ = $\mathcal{E}$-CC-RPC-TP-UW. Consider the two sets $A_{\mathcal{E}} = \{(C, V, p) \mid p \in C$ and p is a unique winner of $\mathcal{E}$ election $(C, V)\}$ and $B_{\mathcal{E}} = \{(C, V, p) \mid p \in C$ and p is not a unique winner of $\mathcal{E}$ election $(C, V)\}$. These sets form a partition of $Y = \{(C, V, p) \mid p \in C\}$, so of course $Y = A_{\mathcal{E}} \cup B_{\mathcal{E}}$. Clearly we also have that $\mathcal{E}$-$\mathcal{T}_1 \subseteq Y$ and that $\mathcal{E}$-$\mathcal{T}_2 \subseteq Y$. We will show that $\mathcal{E}$-$\mathcal{T}_1 = A_{\mathcal{E}} = \mathcal{E}$-$\mathcal{T}_2$. Hemaspaandra et al. [17] show that any election system that satisfies Unique-WARP is, under the TP tie-handling rule in the unique-winner model, immune to constructive control by both run-off partition of candidates and partition of candidates. Thus $\mathcal{E}$-$\mathcal{T}_1 \cap B_{\mathcal{E}} = \emptyset$ and $\mathcal{E}$-$\mathcal{T}_2 \cap B_{\mathcal{E}} = \emptyset$, and it holds that $\mathcal{E}$-$\mathcal{T}_1 \subseteq A_{\mathcal{E}}$ and $\mathcal{E}$-$\mathcal{T}_2 \subseteq A_{\mathcal{E}}$. Fix $(C, V, p) \in A_{\mathcal{E}}$. Then the partition $(\emptyset, C)$ witnesses that $(C, V, p) \in \mathcal{E}$-$\mathcal{T}_1$ as the final round will simply be (C, V) and we know that since $(C, V, p) \in A_{\mathcal{E}}$, p will be the unique winner of the final round. Additionally, the partition $(\emptyset, C)$ also witnesses that $(C, V, p) \in \mathcal{E}$-$\mathcal{T}_2$ as no one will proceed from subelection $(\emptyset, V)$, and only p will proceed from subelection (C, V) (since $(C, V, p) \in A_{\mathcal{E}}$), and p must be the unique winner of $(\{p\}, V)$, the final round (since $\mathcal{E}$ satisfies Unique-WARP and p is the unique winner of $\mathcal{E}$ election (C, V)). ❑

We can build on Thm. 10 to get additional containments.

Corollary 11. *Let $\mathcal{E}$ be an election system that satisfies Unique-WARP. Then, for each $\mathcal{T} \in$* {CC-PC-TE-UW, CC-PC-TE-NUW, CC-RPC-TE-UW, CC-RPC-TE-NUW, CC-PV-TE-UW, CC-PV-TE-NUW, CC-PV-TP-UW, CC-PV-TP-NUW}*, it holds that* $\mathcal{E}$-CC-PC-TP-UW $\subseteq \mathcal{E}$-$\mathcal{T}$ *(equivalently,* $\mathcal{E}$-CC-RPC-TP-UW $\subseteq \mathcal{E}$-$\mathcal{T}$*).*

Since approval voting satisfies Unique-WARP [17], and (clearly) WARP, the above theorems and corollaries apply to approval voting. For each case where only the containment (and not the collapse) is shown, the containment can be made strict under approval voting. See Table 9 for the separation witnesses.

Corollary 12.
(1) Approval-DC-PC-TP-UW = Approval-DC-PC-TE-UW = Approval-DC-RPC-TE-UW = Approval-DC-RPC-TE-NUW = Approval-DC-PC-TE-NUW.
(2) Approval-DC-RPC-TP-NUW = Approval-DC-PC-TP-NUW.
(3) Approval-DC-DC-UW $\subsetneq$ Approval-DC-DV-UW.
(4) Approval-DC-DC-NUW $\subsetneq$ Approval-DC-DV-NUW.

[5] In the unique-winner model, we say an election system is immune to a particular type of control if the given type of control can never change a candidate from not uniquely winning to uniquely winning (if the control type is constructive) or change a candidate from uniquely winning to not uniquely winning (if the control type is destructive) (Hemaspaandra et al. [17], which clarified a slightly flawed immunity definition of Bartholdi et al. [3]). In the nonunique-winner model, we say an election system is immune to a particular type of control if the given type of control can never change a nonwinner to a winner (if the control type is constructive) or change a winner to a nonwinner (if the control type is destructive).

[6] Theorem 7 certainly could also include $\mathcal{E}$-DC-RPC-TE-UW in the equality. However, including that in the theorem would make little sense, since $\mathcal{E}$-DC-RPC-TE-UW = $\mathcal{E}$-DC-PC-TE-UW does not rely on Unique-WARP, but in fact holds for *all* election systems in light of Hemaspaandra et al. [16] and our comment in the paragraph before Corollary 6.

(5) Approval-DC-DC-NUW $\subsetneq$ Approval-DC-DV-UW.
(6) Approval-CC-PC-TP-UW = Approval-CC-RPC-TP-UW.
(7) For each $\mathcal{T} \in$ {CC-PC-TE-UW, CC-PC-TE-NUW, CC-RPC-TE-UW, CC-RPC-TE-NUW, CC-PV-TE-UW, CC-PV-TE-NUW, CC-PV-TP-UW, CC-PV-TP-NUW}*, it holds that* Approval-CC-PC-TP-UW $\subsetneq$ Approval-$\mathcal{T}$*.*

We extend the 5-type collapse in Corollary 12 to a 6-type collapse, and prove more collapses using immunity.

Theorem 13. Approval-DC-RPC-TP-UW = Approval-DC-PC-TP-UW.

Theorem 14. *Approval is immune with respect to* CC-RPC-TP-NUW *and* CC-PC-TP-NUW.

Theorem 15. Approval-CC-PC-TP-NUW = Approval-CC-RPC-TP-NUW.

Corollary 16. *For each* $\mathcal{T} \in$ {CC-PC-TE-NUW, CC-RPC-TE-NUW, CC-PV-TP-NUW}*, it holds that* Approval-CC-PC-TP-NUW $\subsetneq$ Approval-$\mathcal{T}$*.*

Our remaining results use direct arguments.

Theorem 17. Approval-DC-PV-TE-UW = Approval-DC-PV-TE-NUW.

Theorem 18. Approval-CC-PC-TE-NUW = Approval-CC-RPC-TE-NUW.

Proof. The approach we take is a bit more tedious than would be needed to just prove Theorem 18, as we have structured the proof to also establish Corollary 19.

$\subseteq$: Let $(C, V, p) \in$ Approval-CC-PC-TE-NUW and let (C_1, C_2) be a candidate partition that witnesses this membership. Consider the case where p uniquely wins the final round. If $p \in C_1$, then p uniquely wins (C_1, V) and in the final round defeats all candidates in C_2. If $p \in C_2$, then in the final round p defeats the candidate (if any) that survives the TE tie-handling rule regarding the subelection (C_1, V) as well as all the candidates in $C_2 - \{p\}$. Regardless of which case holds, the partition (C_1, C_2) will witness $(C, V, p) \in$ Approval-CC-RPC-TE-NUW since p will uniquely win its subelection and then will defeat any candidate that moves forward from the other subelection. If p does not uniquely win the final round, then there is at least one other candidate that ties with p in the final round. If $p \in C_1$, then p must uniquely win in (C_1, V) and as (since (C_1, C_2) witnesses $(C, V, p) \in$ Approval-CC-PC-TE-NUW) no candidate in C_2 can have a score greater than p's, the partition (C_1, C_2) suffices to witness $(C, V, p) \in$ Approval-CC-RPC-TE-NUW. If $p \in C_2$, then under partition (C_1, C_2) p could first-round tie with a candidate and be eliminated (under run-off partition of candidates due to the TE rule). However, let T denote the (possibly empty) set of candidates (other than p) that tie with p in (C_2, V). Then the partition $(C_1 \cup T, C_2 - T)$ witnesses $(C, V, p) \in$ Approval-CC-RPC-TE-NUW, since p will uniquely win $(C_2 - T, V)$ and will either tie or defeat the winner (if any) of $(C_1 \cup T, V)$.

$\supseteq$: Let $(C, V, p) \in$ Approval-CC-RPC-TE-NUW and let (C_1, C_2) be a candidate partition that witnesses this membership. Without loss of generality, assume that $p \in C_1$. Thus it holds that p uniquely wins (C_1, V). If p uniquely wins the final round, then p also defeats the candidate (if any) that moves forward from (C_2, V). Thus the partition (C_2, C_1) will also witness $(C, V, p) \in$ Approval-CC-PC-TE-NUW (since p has strictly more approvals than any candidate other than itself). If p does not uniquely win the final round, then there is another candidate d, who is the unique winner of (C_2, V) and ties with p in the final round. Again the partition (C_2, C_1) suffices to witness $(C, V, p) \in$ Approval-CC-PC-TE-NUW since d proceeds to the final round and both p and d win there due to their numbers of approvals. ❑

Corollary (to the Proof) 19. Approval-CC-PC-TE-UW = Approval-CC-RPC-TE-UW.

Proof. This is an immediate corollary to the above proof of Theorem 18, as the proof was intentionally structured to ensure that if the witness of one type made p a unique winner of the final round, then the constructed-above (sometimes different) partition for the other type also made p a unique winner of the final round under that other type of control. ❑

Theorem 20. Approval-DC-PV-TP-UW $\subsetneq$ Approval-DC-PV-TE-NUW.

Theorem 21 and Corollary 22 are surprising, since they are about partitioning different components of elections.

Theorem 21. Approval-DC-RPC-TE-NUW $\subsetneq$ Approval-DC-PV-TP-UW.

Corollary 22. Approval-DC-RPC-TE-NUW $\subsetneq$ Approval-DC-PV-TE-NUW.

4 RELATED WORK

Bartholdi et al. [3] introduced and studied the goal of making a particular candidate be an untied winner of the election (constructive control in the unique-winner model). Hemaspaandra et al. [17] introduced "destructive control" versions of each constructive control type: The goal is to prevent (via the given control action) a focus candidate from becoming a unique winner. Hemaspaandra et al. [16, Footnote 5] argue that the nonunique-winner model is a better model to study than the unique-winner model. That paper—with its 7 control-pair collapses and its one separation ($\mathcal{E}$-DC-RPC-TP-UW $\neq$ $\mathcal{E}$-DC-PC-TP-UW, though their $\mathcal{E}$ is not (candidate-)neutral)—is the paper most related to ours.

Hemaspaandra et al. [17] clarified the ways of handling ties in the first-round elections of the (two-round) "partition" control types of Bartholdi, Tovey, and Trick, naming the two approaches "ties promote" and "ties eliminate"; they did this because although tie-handling had previously been suggested as being unimportant, their paper establishes that, for example, for plurality the choice between those two rules spells the difference between being NP-complete and belonging to P. The "adding candidates" control attack of Bartholdi, et al., was anomalously defined, and Hemaspaandra et al. [18] kept the original notion but renamed it "unlimited adding of candidates," and introduced, under the thus-available name "adding candidates," the version that is analogous to the other Bartholdi, et al., add/delete types, and the subsequent papers have followed that notational shift.

Altogether, the Bartholdi, Tovey, and Trick control attack set, under the above clarifications and enrichments, yields a set of

Table 1: Summary of separations and collapses. Blue indicates results due to or inherited from Hemaspaandra et al. [16]. Red indicates results due to the present paper.

Election System	Set Classification			Subclassification of Separations		
	Separations	Collapses	Open	"$\subsetneq$"/"$\supsetneq$"	Incomparable	Open
General Case	1 + 314 = 315†	7	0	38	277	0
Plurality	315	7	0	38	277	0
Veto	314	7 + 1 = 8	0	58	256	0
Approval Voting	301	7 + 14 = 21	0	88	213	0

† Or 0 + 315 for the pure social choice approach to candidate names (see Footnote 2).

eleven constructive control attacks and eleven destructive control attacks. As mentioned earlier, this in some sense forms a "standard" benchmark set of attacks (though some papers use subsets of that collection, and other papers have taken control in additional directions, e.g., resolute control [15, 26]). For example, the excellent survey chapter on control and bribery by Faliszewski and Rothe [14] uses precisely those 22 control attacks, as does the recent paper on search versus decision of Hemaspaandra et al. [16]. Since the field has not yet resolved whether nonunique-winner or unique-winner is the right standard—indeed, the two just-cited sources make different choices—and as discovering cases when a control type in one of those models turns out to be identical to a control type in the other is itself interesting, this paper has covered both models, and thus $2 \cdot (11 + 11) = 44$ control types.

The additional, different control types known as resolute control [15, 26] are quite interesting. Resolute control asks whether there is some action (from a certain range of actions) that keeps every one of a certain collection of candidates from being a winner. This might seem to be the same as our function model for the case of (nonunique-winner) destructive control, but it is not. In our function model for nonunique-winner destructive control, we are speaking of the collection of candidates who can individually be prevented from winning by some control action. But even if two or more candidates belong to our function's output, they could be put into that output by different control actions, and there might be no single action that blocks both simultaneously. In brief, resolute control is focused on blocking whole groups, and our function model in contrast is a refinement of set separations and focuses on individual candidates to let us identify new containment patterns between control types.

Collapsing or separating control types is not *directly* about complexity. However, doing so is highly relevant to complexity, as the types were defined as natural benchmarks whose complexity could be studied. In fact, there has been something of a race to find natural systems in which very many control attacks are NP-hard (see Additional Note 5). Among the many systems that have done well in that race are, along with some of the key papers that analyzed their complexity, Schulze elections [22, 24], ranked-pair elections [24], SP-AV elections [9], normalized range voting [21], fallback elections [10, 12] (see also [8]), and Bucklin elections [10] (see also [8]). Faliszewski and Rothe [14, Table 7.3] provide a nice table, for the 22 unique-winner control cases, of what is known for 12 voting systems. As to the three important systems spotlighted in our paper, plurality's control was explored by Bartholdi et al. [3] and Hemaspaandra et al. [17], veto has been studied by Lin [19] and Maushagen and Rothe [20] (see also Table 1 of [11]), and approval has been studied by Hemaspaandra et al. [17] (see also [4]).

5 CONCLUSIONS AND OPEN PROBLEMS

Table 1 summarizes our results. We established that in the general (universally quantified) case there are no collapsing pairs (by which we always mean among the standard 44 control types) other than the 7 collapsing pairs identified by Hemaspaandra et al. [16], and that plurality has no collapsing pairs beyond those 7. For veto and approval voting we discovered additional collapsing pairs beyond those inherited 7, but we also established that veto and approval voting, after our work, have no remaining undiscovered collapsing pairs.

Our work helps clarify the landscape of which control pairs do or do not collapse, both in the general (universally quantified) case, and for plurality, veto, and approval voting.

We also refined all our separations, and in doing so uncovered containment relationships—including many that do not follow from the relationship between the nonunique-winner model and the unique-winner model.

A number of interesting open directions are suggested by our work. One is, for important concrete election systems other than plurality, veto, and approval voting, to completely classify the collapses and separations that hold for those specific systems. Another direction—building on the results using Unique-WARP—is to find sufficient conditions (or, better still, necessary-and-sufficient conditions) for many control-pair collapses in terms of axiomatic properties of election systems. Though our goal for separations was to subclassify each separation into exactly one of the three cases "$\subsetneq$", "$\supsetneq$", or incomparability, in our tables we have also noted those cases where our constructions establish strong incomparability; one could for the cases where we list incomparability try to establish strong incomparability. An additional open direction is to see whether already-studied control types beyond the 44 investigated here collapse with each other or with some of the 44, either in general or for important concrete election systems.

Finally, control types are defined as sets. When a pair of control types collapses, those sets are equal and thus certainly are of the same complexity. However, it would be interesting to see whether for collapsing control types, their complexities as *search* problems are or are not polynomially related. That issue, inspired by the work of Hemaspaandra et al. [16] and an earlier version of the present paper, has recently been studied by Carleton et al. [5, 7].

ACKNOWLEDGMENTS

This work was supported in part by NSF grant CCF-2006496 and CIFellows grant CIF2020-UR-36. We thank the anonymous referees for their helpful comments and suggestions.

REFERENCES

[1] John J. Bartholdi, III, Craig A. Tovey, and Michael A. Trick. 1989. The Computational Difficulty of Manipulating an Election. *Social Choice and Welfare* 6, 3 (1989), 227–241.

[2] John J. Bartholdi, III, Craig A. Tovey, and Michael A. Trick. 1989. Voting Schemes for Which it Can Be Difficult to Tell Who Won The Election. *Social Choice and Welfare* 6, 2 (1989), 157–165.

[3] John J. Bartholdi, III, Craig A. Tovey, and Michael A. Trick. 1992. How Hard is it to Control an Election? *Mathematical and Computer Modeling* 16, 8–9 (1992), 27–40.

[4] Dorothea Baumeister, Gábor Erdélyi, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. 2010. Computational Aspects of Approval Voting. In *Handbook on Approval Voting*, Jean-François Laslier and M. Remzi Sanver (Eds.). Springer, 199–251.

[5] Benjamin Carleton, Michael C. Chavrimootoo, Lane A. Hemaspaandra, David E. Narváez, Conor Taliancich, and Henry B. Welles. 2022. *Search versus Search for Collapsing Electoral Control Types.* Technical Report arXiv:2207.03049 [cs.GT]. Computing Research Repository, arXiv.org/corr/. Revised, October 2022.

[6] Benjamin Carleton, Michael C. Chavrimootoo, Lane A. Hemaspaandra, David E. Narváez, Conor Taliancich, and Henry B. Welles. 2022. *Separating and Collapsing Electoral Control Types.* Technical Report arXiv:2207.00710 [cs.MA]. Computing Research Repository, arXiv.org/corr/. Revised, February 2023.

[7] Benjamin Carleton, Michael C. Chavrimootoo, Lane A. Hemaspaandra, David E. Narváez, Conor Taliancich, and Henry B. Welles. 2023. Search versus Search for Collapsing Electoral Control Types (Extended Abstract). *In these proceedings.*

[8] Gábor Erdélyi, Michael R. Fellows, Jörg Rothe, and Lena Schend. 2015. Control Complexity in Bucklin and Fallback Voting: A Theoretical Analysis. *J. Comput. System Sci.* 81, 4 (2015), 632–660.

[9] Gábor Erdélyi, Markus Nowak, and Jörg Rothe. 2009. Sincere-Strategy Preference-Based Approval Voting Fully Resists Constructive Control and Broadly Resists Destructive Control. *Mathematical Logic Quarterly* 55, 4 (2009), 425–443.

[10] Gábor Erdélyi, Lena Piras, and Jörg Rothe. 2011. The Complexity of Voter Partition in Bucklin and Fallback Voting: Solving Three Open Problems. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, 837–844.

[11] Gábor Erdélyi, Christian Reger, and Yongjie Yang. 2019. Towards Completing the Puzzle: Solving Open Problems for Control in Elections. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, 846–854.

[12] Gábor Erdélyi and Jörg Rothe. 2010. Control Complexity in Fallback Voting. In *Proceedings the 16th Australasian Theory Symposium.* Australian Computer Society, 39–48.

[13] Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. 2009. Llull and Copeland Voting Computationally Resist Bribery and Constructive Control. *Journal of Artificial Intelligence Research* 35 (2009), 275–341.

[14] Piotr Faliszewski and Jörg Rothe. 2016. Control and Bribery in Voting. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, 146–168.

[15] Sushmita Gupta, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. 2022. Resolute control: Forbidding candidates from winning an election is hard. *Theoretical Computer Science* 915 (2022), 74–89.

[16] Edith Hemaspaandra, Lane A. Hemaspaandra, and Curtis Menton. 2020. Search versus Decision for Election Manipulation Problems. *ACM Transactions on Computation Theory* 12, #1, Article 3 (2020), 1–42.

[17] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. 2007. Anyone but Him: The Complexity of Precluding an Alternative. *Artificial Intelligence* 171, 5–6 (2007), 255–285.

[18] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. 2009. Hybrid Elections Broaden Complexity-Theoretic Resistance to Control. *Mathematical Logic Quarterly* 55, 4 (2009), 397–424.

[19] Andrew Lin. 2012. *Solving Hard Problems in Election Systems.* Ph.D. Dissertation. Rochester Institute of Technology, Rochester, NY.

[20] Cynthia Maushagen and Jörg Rothe. 2018. Complexity of Control by Partitioning Veto Elections and of Control by Adding Candidates to Plurality Elections. *Annals of Mathematics and Artificial Intelligence* 82, 4 (2018), 219–244.

[21] Curtis Menton. 2013. Normalized Range Voting Broadly Resists Control. *Theory of Computing Systems* 53, 4 (2013), 507–531.

[22] Curtis Menton and Preetjot Singh. 2013. Control Complexity of Schulze Voting. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence.* AAAI Press, 286–292.

[23] Marc Neveling and Jörg Rothe. 2017. Solving Seven Open Problems of Offline and Online Control in Borda Elections. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence.* AAAI Press, 3029–3035.

[24] David C. Parkes and Lirong Xia. 2012. A Complexity-of-Strategic-Behavior Comparison between Schulze's Rule and Ranked Pairs. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence.* AAAI Press, 1429–1435.

[25] Paul A. Samuelson. 1938. A Note on the Pure Theory of Consumer's Behaviour. *Economica* 5, 17 (1938), 61–71.

[26] Yongjie Yang and Jianxin Wang. 2017. Anyone But Them: The Complexity Challenge for a Resolute Election Controller. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, 1133–1141.