Hybrid Elections Broaden Complexity-Theoretic Resistance to Control[1]

Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe

Abstract

Electoral control refers to attempts by an election's organizer ("the chair") to influence the outcome by adding/deleting/partitioning voters or candidates. The groundbreaking work of Bartholdi, Tovey, and Trick [BTT92] on (constructive) control proposes computational complexity as a means of resisting control attempts: Look for election systems where the chair's task in seeking control is itself computationally infeasible.

We introduce and study a method of combining two or more candidate-anonymous election schemes in such a way that the combined scheme possesses all the resistances to control (i.e., all the NP-hardnesses of control) possessed by *any* of its constituents: It combines their strengths. From this and new resistance constructions, we prove for the first time that there exists an election scheme that is resistant to all twenty standard types of electoral control.

Key words: multiagent systems, computational social choice, preference aggregation, computational complexity, electoral control.

1 Introduction

Elections are a way of, from a collection of voters' (or agents') individual preferences over candidates (or alternatives), selecting a winner (or outcome). The importance of and study of elections is obviously central in political science, but also spans such fields as economics, mathematics, operations research, and computer science. Within computer science, the applications of elections are most prominent in distributed AI, most particularly in the study of multiagent systems. For example, voting has been concretely proposed as a computational mechanism for planning [ER91,ER93] and has also been suggested as an approach to collaborative filtering [PHG00]. However, voting also has received attention within the study of systems. After all, many distributed algorithms must start by selecting a leader, and election techniques have also been proposed to attack the web page rank aggregation problem and the related issue of lessening the spam level of results from web searches [DKNS01,FKS03]. Indeed, in these days of a massive internet with many pages, many surfers, and many robots, of intracorporate decision-making potentially involving electronic input

[1]Conference version to appear in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*; a full version is available as [HHR06]. Supported in part by DFG grants RO 1202/9-1 and RO 1202/9-3, NSF grants CCR-0311021 and CCF-0426761, a Friedrich Wilhelm Bessel Research Award, and the Alexander von Humboldt Foundation's TransCoop program.

from many units/individuals/warehouses/trucks/sources, and more generally of massive computational settings including many actors, it is easy to note any number of situations in which elections are natural and in which the number of candidates and/or voters might be massive. For example, suppose amazon.com were to select a "page of the week" via an election where the candidates were all its web pages and the voters were all visiting surfers (with preferences inferred from their page-viewing times or patterns); such an election would have an enormous number of candidates and voters. All these applications are exciting, but immediately bring to a theoretician's mind the worry of whether the complexity of implementing election systems is satisfyingly low and whether the complexity of distorting (controlling or manipulating) election systems is reassuringly high.

Since the complexity of elections is a topic whose importance has made itself clear, it is natural to ask whether the standard tools and techniques of complexity-theoretic study exist in the context of elections. One important technique in complexity is the combination of problems. For example, for sets in complexity theory, a standard approach to combination is the join (also known as the disjoint union and as the marked union): $A \oplus B = \{0x \mid x \in A\} \cup \{1y \mid y \in B\}$.

In some sense, our work in this paper can be thought of as simply providing, for elections, an analog of the join. That is, we will propose a method of combining two (or more) elections in a way that will maintain desirable simplicity properties (e.g., if all of the constituent elections have polynomial-time winner algorithms then so will our combined election) while also inheriting quite aggressively desirable hardness properties (we will show that any resistance-to-control—in the sense that is standard [BTT92] and that we will provide a definition of later—possessed by even one of the constituent elections will be possessed by the combined election). One cannot directly use a join to achieve this, because the join of two sets modeling elections is not itself an election. Rather, we must find a way of embedding into election specifications—lists of voter preferences over candidates—triggers that both allow us to embed and switch between all the underlying election systems and to not have such switching go uncontrollably haywire when faced with electoral distortions such as adding/deleting/partitioning voters/candidates, since we wish hardness with respect to control by such mechanisms to be preserved.

We above have phrased this paper's theme as the development of a way of combining multiple election systems—and in doing so, have desirable types of simplicity/complexity inheritance. However, this paper also has in mind a very specific application—both for its own interest and as a sounding board against which our election hybridization scheme can be tested. This application is the control of election systems.

In election control, we ask whether an election's organizer (the chair) can by some specific type of manipulation of the election's structure (adding/deleting/partitioning voters/candidates) cause a specified candidate to be the (unique) winner. As mentioned earlier, the complexity-theoretic study of control was proposed by Bartholdi, Tovey, and Trick in 1992 [BTT92]. We will closely follow their model. In this model, the chair is assumed to have knowledge of the vote that will be cast by each voter, and there are ten different types of control (candidate addition, candidate deletion, voter addition, voter deletion, partition of candidates, run-off partition of candidates, and partition of voters [BTT92]—and for each of the three partition cases one can have subelection ties promote or can have subelection ties eliminate, see [HHR05a]).

Of course, the dream case would be to find an election system that has the desirable property of having a polynomial-time algorithm for evaluating who won, but that also has the property that for every single one of the ten standard types of control it is computationally infeasible (NP-hard) to assert such control. Unfortunately, no system yet has been proven resistant to all ten types of control. In fact, given that broad "impossibility" results exist for niceness of preference aggregation systems (e.g., Arrow's Theorem [Arr63]) and for nonmanipulability of election systems (e.g., the Gibbard–Satterthwaite and Duggan–Schwartz Theorems ([Gib73,Sat75,DS00], see also [Tay05])), one might even momentarily wonder whether the "dream case" mentioned above can be proven impossible via proving a theorem of the following form: "For no election system whose winner complexity is in P are all ten types of control NP-hard." However, such a claim is proven impossible by our work: Our hybrid system in fact will allow us to combine *all* the resistance types of the underlying elections. And while doing so, it will preserve the winner-evaluation simplicity of the underlying elections. Thus, in particular, we conclude that the "dream case" holds: There is an election system—namely, our hybridization of plurality and Condorcet elections—that is resistant to all ten types of constructive control. We also show—by building some artificial election systems achieving resistance to destructive control types for which no system has been previously proven resistant and then invoking our hybridization machinery—that there is an election system that is resistant to all ten types of destructive control (in which the chair's goal is to preclude a given candidate from being the (unique) winner) as well as to all ten types of constructive control (Theorem 3.8).

Our hybridization system takes multiple elections and maintains their simplicity while inheriting each resistance-to-control possessed by any one of its constituents. Thus, it in effect unions together all their resistances—thus the "broaden" of our title. We mention in passing that in the quite different setting of election manipulation (which regards not actions by the chair but rather which regards voters altering their preferences in an attempt to influence who becomes the winner) [BTT89a], there has been some work by Conitzer and Sandholm [CS03] regarding making manipulation hard, even for systems where it is not hard, by changing the system by going to a two-stage election in which a single elimination preround is added, and Elkind and Lipmaa [EL05] have generalized this to a sequence of elimination rounds conducted under some system(s) followed by an election under some other system. Though the latter paper like this paper uses the term "hybrid," the domains differ sharply and the methods of election combination are nearly opposite: Our approach (in order to broaden resistance to control) embeds the election systems in parallel and theirs (in order to fight manipulation) strings them out in sequence. Of the two approaches, ours far more strongly has the flavor of our simple motivating example, the join.

The previous work most closely related to that of this paper is the constructive control work of Bartholdi, Tovey, and Trick [BTT92] and the destructive control work of Hemaspaandra, Hemaspaandra, and Rothe [HHR05a]. Work on bribery is somewhat related to this paper, in the sense that bribery can be viewed as sharing aspects of both manipulation and control [FHH06]. Of course, all the classical [BTT89b,BTT89a,BO91] and recent papers (of which we particularly point out, for its broad framework and generality, the work of Spakowski and Vogel [SV00]) on the complexity of election problems share this paper's goal of better understanding the relationship between complexity and elections.

We here omit proofs due to lack of space, but detailed proofs are available in the full version of this paper [HHR06].

2 Definitions and Discussion

2.1 Elections

An election system (or election rule or election scheme or voting system) $\mathcal{E}$ is simply a mapping from (finite though arbitrary-sized) sets (actually, mathematically, they are multisets) V of votes (each a preference order—strict, transitive, and complete—over a finite candidate set) to (possibly empty, possibly nonstrict) subsets of the candidates. All votes in a given V are over the same candidate set, but different V's of course can be over different (finite) candidate sets. Each candidate that for a given set of votes is in $\mathcal{E}$'s output is said to be a *winner*. If for a given input $\mathcal{E}$ outputs a set of cardinality one, that candidate is said to be the *unique winner*. Election control focuses on making candidates be unique winners and on precluding them from being unique winners.

Throughout this paper, a voter's preference order will be exactly that: a tie-free linear order over the candidates. And we will discuss and hybridize only election systems based on preference orders.

We now define two common election systems, plurality voting and Condorcet voting. In *plurality voting*, the winners are the candidates who are ranked first the most. In *Condorcet voting*, the winners are all candidates (note: there can be at most one and there might be zero) who strictly beat each other candidate in head-on-head majority-rule elections (i.e., get *strictly* more than half the votes in each such election). For widely used systems such as plurality voting, we will write plurality rather than $\mathcal{E}_{\text{plurality}}$.

We say that an election system $\mathcal{E}$ is *candidate-anonymous* if for every pair of sets of votes V and V', $\|V\| = \|V'\|$, such that V' can be created from V by applying some one-to-one mapping h from the candidate names in V onto new candidate names in V' (e.g., each instance of "George" in V is mapped by h

to "John" in V' and each instance of "John" in V is mapped by h to "Hillary" in V' and each instance of "Ralph" in V is mapped by h to "Ralph" in V') it holds that $\mathcal{E}(V') = \{c' \mid (\exists c \in \mathcal{E}(V))\,[h(c) = c']\}$. Informally put, candidate-anonymity says that the strings we may use to name the candidates are all created equal. Note that most natural systems are candidate-anonymous. For example, both the election systems mentioned immediately above—plurality-rule elections and the election system of Condorcet—are candidate-anonymous.

2.2 Our Hybridization Scheme

We now define our basic hybridization scheme, *hybrid*.

Definition 2.1 *Let $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1}$ be election rules that take as input voters' preference orders. Define hybrid$(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ to be the election rule that does the following: If there is at least one candidate and all candidate names (viewed as natural numbers via the standard bijection between Σ^* and $\mathbb{N}$) are congruent, modulo k, to i (for some i, $0 \leq i \leq k-1$) then use election rule $\mathcal{E}_i$. Otherwise use, by convention, $\mathcal{E}_{k-1}$ as the default election rule.*

Having defined our system there is much to discuss. Why did we choose this system? What are its properties? What other approaches did we choose not to use, and why? What aspects of the input is our method for switching between election systems using, and what aspects is it choosing not to exploit, and what are the costs associated with our choices?

As to the properties of this system, Section 3 is devoted to that, but most crucially we will see that this system possesses every resistance-to-control property possessed by even one of its constituents. And this will hold essentially due to the fact that *hybrid* is a close analog of the effect of a join: It splices the constituents together in such a way that key questions about the constituent systems can easily be many-one polynomial-time reduced ($\leq_m^p$-reduced or reduced, for short) to questions about their hybrid.

As to why we chose this particular system, note that *hybrid* "switches" between constituent systems via wildly redundant information. This will let us keep deletions/partitions of voters/candidates from causing a switch between the underlying systems (if the starting state routed us to a nondefault case). Note that some other approaches that one might take are more sensitive to deletions. For example, suppose we wanted to hybridize just two election systems and decided to do so by using the first election system exactly if the first voter's most disliked candidate's name is lexicographically less than the first voter's second-most-disliked candidate's name. Note that if, as part of our control problem, that voter is deleted, that might suddenly change the system to which the problem is routed. Or, as another example, if we use the modulo k value of the name of the lexicographically smallest candidate to control switching between the k election systems, then that hybridization approach would be very sensitive to jumping between systems if, as part of our control problem, that candidate is deleted. These examples give some idea of why we chose the approach we did, though admittedly even it can in some cases be nudged into jumping between systems—but at least this happens in very limited, very crisply delineated cases and in ways that we will generally be able to appropriately handle.

Finally, we come to what we allow ourselves to use to control the switching, what we choose not to use, and what price we pay for our choices. What we use (as is allowed in the [BTT92] model) are the candidates' names and only the candidates' names. We use absolutely nothing else to control switching between elections. We do not use voters' names. Indeed, in the [BTT92] model that we follow, voters (unlike candidates) do not even have names. But since the votes are input as a list, their ordering itself could be used to pass bits of information—e.g., we could look at whether the first vote in the list viewed as a string is lexicographically less than the last vote in the list viewed as a string. We in no way "cheat" by exploiting such input-order information, either for the votes or for the list of candidates (as per [BTT92], formally the candidate set is passed in separately to cover a certain boundary case). Our "switch" is based purely on candidates' names and just candidates' names. This also points to the price we pay for this choice: Even when all its constituent elections are candidate-anonymous, *hybrid* may not possess candidate-anonymity.

2.3 Types of Constructive and Destructive Control

Constructive control problems ask whether a certain class of actions by the election's chair can make a specified candidate the election's unique winner. Constructive control was first defined and studied by Bartholdi, Tovey, and Trick [BTT92]. Destructive control problems ask whether a certain class of actions by the election's chair can make a specified candidate fail to be a unique winner of the election. Destructive control was defined and studied by Hemaspaandra, Hemaspaandra, and Rothe [HHR05a], and in the different context of electoral manipulation destruction was introduced even earlier by Conitzer, Lang, and Sandholm [CS02,CLS03].

Bartholdi, Tovey, and Trick's [BTT92] groundbreaking paper defined seven types of electoral control. Among those seven, three are partition problems for which there are two different natural approaches to handling ties in subelections (see [HHR05a] which introduced these tie-handling models for this context): eliminating tied subelection winners (the "TE" model) or promoting tied subelection winners (the "TP" model). Thus, there are $(7-3)+2\cdot 3=10$ different standard types of constructive control, and there are essentially the same ten types of destructive control.

Since it is exceedingly important to not use a slightly different problem statement than earlier work whose results we will be drawing on, we will state the seven standard constructive control types (which become ten with the three partition control types each having both "TE" and "TP" versions) and their destructive analogs using word-for-word definitions from [HHR05a,HHR05b], which themselves are based closely and often identically on [BTT92] (see the discussion in [HHR05a,HHR05b]).

Though V, the set of votes, is conceptually a multiset as in the previous related work, we take the view that the votes are input as a list ("the ballots"), and in particular are not directly input as a multiset in which cardinalities are input in binary (though we will mention later that our main result about *hybrid* holds also in that quite different model).

Constructive (Destructive) Control by Adding Candidates: Given a set C of qualified candidates and a distinguished candidate $c \in C$, a set D of possible spoiler candidates, and a set V of voters with preferences over $C \cup D$, is there a choice of candidates from D whose entry into the election would assure that c is (not) the unique winner?

Constructive (Destructive) Control by Deleting Candidates: Given a set C of candidates, a distinguished candidate $c \in C$, a set V of voters, and a positive integer $k < \|C\|$, is there a set of k or fewer candidates in C whose disqualification would assure that c is (not) the unique winner?

Constructive (Destructive) Control by Partition of Candidates: Given a set C of candidates, a distinguished candidate $c \in C$, and a set V of voters, is there a partition of C into C_1 and C_2 such that c is (not) the unique winner in the sequential two-stage election in which the winners in the subelection (C_1, V) who survive the tie-handling rule move forward to face the candidates in C_2 (with voter set V)?

Constructive (Destructive) Control by Run-Off Partition of Candidates: Given a set C of candidates, a distinguished candidate $c \in C$, and a set V of voters, is there a partition of C into C_1 and C_2 such that c is (not) the unique winner of the election in which those candidates surviving (with respect to the tie-handling rule) subelections (C_1, V) and (C_2, V) have a run-off with voter set V?

Constructive (Destructive) Control by Adding Voters: Given a set of candidates C and a distinguished candidate $c \in C$, a set V of registered voters, an additional set W of yet unregistered voters (both V and W have preferences over C), and a positive integer $k \leq \|W\|$, is there a set of k or fewer voters from W whose registration would assure that c is (not) the unique winner?

Constructive (Destructive) Control by Deleting Voters: Given a set of candidates C, a distinguished candidate $c \in C$, a set V of voters, and a positive integer $k \leq \|V\|$, is there a set of k or fewer voters in V whose disenfranchisement would assure that c is (not) the unique winner?

Constructive (Destructive) Control by Partition of Voters: Given a set of candidates C, a distinguished candidate $c \in C$, and a set V of voters, is there a partition of V into V_1 and V_2 such that c is (not) the unique winner in the hierarchical two-stage election in which the survivors of (C, V_1) and (C, V_2) run against each other with voter set V?

2.4 Immunity, Susceptibility, Vulnerability, Resistance

Again, to allow consistency with earlier papers and their results, we take this definition from [HHR05a,HHR05b], with the important exception regarding resistance discussed below Definition 2.2. It is worth noting that immunity and susceptibility both are "directional" (can we change *this*?) but that vulnerability and resistance are, in contrast, outcome-oriented (can we end up with *this* happening?) and complexity-focused.

Definition 2.2 *We say that a voting system is* immune to control *in a given model of control (e.g., "destructive control via adding candidates") if the model regards constructive control and it is never possible for the chair to by using his/her allowed model of control change a given candidate from being not a unique winner to being the unique winner, or the model regards destructive control and it is never possible for the chair to by using his/her allowed model of control change a given candidate from being the unique winner to not being a unique winner. If a system is not immune to a type of control, it is said to be* susceptible *to that type of control.*

A voting system is said to be (computationally) vulnerable to control *if it is susceptible to control and the corresponding language problem is computationally easy (i.e., solvable in polynomial time).*

A voting system is said to be resistant to control *if it is susceptible to control but the corresponding language problem is computationally hard (i.e.,* NP*-hard).*

We have diverged from all previous papers by defining resistance as meaning NP-hardness (i.e., NP-$\leq_m^p$-hardness) rather than NP-completeness (i.e., NP-$\leq_m^p$-completeness). In [BTT92], where the notion was defined, all problems were trivially in NP. But control problems might in difficulty exceed NP-completeness, and so the notion of resistance is better captured by NP-hardness.

An anonymous IJCAI referee commented that even polynomial-time algorithms can be expensive to run on sufficiently large inputs. We mention that though the comment is correct, almost any would-be controller would probably much prefer that challenge, solving a P problem on large inputs, to the challenge our results give him/her, namely, solving an NP-complete problem on large inputs. We also mention that since the hybrid scheme is designed so as to inherit resistances from the underlying schemes, if a hybrid requires extreme ratios between the number of candidates and the number of voters to display asymptotic hardness, that is purely due to inheriting that from the underlying systems. Indeed, if anything the hybrid is less likely to show that behavior since, informally put, if even one of the underlying systems achieves asymptotic hardness even away from extreme ratios between the number of candidates and the number of voters, then their hybrid will also.

2.5 Inheritance

We will be centrally concerned with the extent to which $hybrid(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ inherits the properties of its constituents. To do so, we formally define our notions of inheritance (if all the constituents have a property then so does their hybrid) and of strong inheritance (if even one of the constituents has a property then so does the hybrid).

Definition 2.3 *We say that a property* Γ *is* strongly inherited *(respectively,* inherited*) by hybrid if the following holds: Let* $k \in \mathbb{N}^+$*. Let* $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1}$ *be candidate-anonymous election systems (each taking as input* (C, V)*, with* V *a list of preference orders). It holds that hybrid*$(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ *has property* Γ *if at least one of its constituents has (respectively, all its constituents have) property* Γ*.*

Definition 2.3 builds in the assumption that all constituents are candidate-anonymous. This assumption isn't overly onerous since as mentioned earlier candidate-anonymity is very common—but will be used in many of our proofs.

Though we will build candidate-anonymity into the assumptions underlying inheritance, we will often try to let interested readers know when that assumption is not needed. In particular, when we say "inherited (and flexibly so)" or "strongly inherited (and flexibly so)," the "(and flexibly so)" indicates that the claim holds even if in Definition 2.3 the words "candidate-anonymous" are deleted. For example, the following easy but quite important claim follows easily from the definition of *hybrid*.

Proposition 2.4 *"Winner problem membership in* P*," "unique winner problem membership in* P*," "winner problem membership in* NP*," and "unique winner problem membership in* NP*" are inherited (and flexibly so) by hybrid.*

3 Inheritance and Hybrid Elections: Results

In this section we will discuss the inheritance properties of *hybrid* with respect to susceptibility, resistance, immunity, and vulnerability. Table 1 summarizes our results for the cases of constructive control and destructive control. (This table does not discuss/include the issue of when "(and flexibly so)" holds, i.e., when the candidate-anonymity assumption is not needed, but rather focuses on our basic inheritance definition.)

3.1 Susceptibility

We first note that susceptibility strongly inherits. We remind the reader that throughout this paper, when we speak of an election system, we always implicitly mean an election system that takes as input (C, V) with V a list of preference orders over C.

Theorem 3.1 *Let* $k \in \mathbb{N}^+$ *and let* $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1}$ *be election systems. Let* Φ *be one of the standard twenty types of (constructive and destructive) control. If for at least one* i*,* $0 \leq i \leq k-1$*,* $\mathcal{E}_i$ *is candidate-anonymous and susceptible to* Φ*, then hybrid*$(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ *is susceptible to* Φ*.*

Control by	Susceptibility	Resistance	Immunity	Vulnerability
Adding Candidates	SI	SI	Not I / I*	I
Deleting Candidates	SI	SI	I / Not I*	I iff P = NP
Partition of Candidates (TE)	SI	SI	Not I	On (⋆⋆) systems: I iff SI iff P = NP
Partition of Candidates (TP)	SI	SI	Not I	On (⋆) systems: I iff SI iff P = NP
Run-off Partition of Candidates (TE)	SI	SI	Not I	On (⋆⋆) systems: I iff SI iff P = NP
Run-off Partition of Candidates (TP)	SI	SI	Not I	On (⋆) systems: I iff SI iff P = NP
Adding Voters	SI	SI	I	I
Deleting Voters	SI	SI	I	I
Partition of Voters (TE)	SI	SI	I	I
Partition of Voters (TP)	SI	SI	I	I

Table 1: Inheritance results that hold or provably fail for *hybrid*. Key: I = Inherits. SI = Strongly Inherits. Boxes without a * state results for both constructive and destructive control. In boxes with a *, the * refers to the destructive control case. "On (⋆) systems" is a shorthand for "On election systems having winner problems in the polynomial hierarchy." "On (⋆⋆) systems" is a shorthand for "On election systems having unique winner problems in the polynomial hierarchy."

Corollary 3.2 *hybrid strongly inherits susceptibility to each of the standard twenty types of control.*

3.2 Resistance

We now come to the most important inheritance case, namely, that of resistance. Since our hope is that hybrid elections will broaden resistance, the ideal case would be to show that resistance is strongly inherited. And we will indeed show that, and from it will conclude that there exist election systems that are resistant to all twenty standard types of control.

We first state the key result, which uses the fact that *hybrid* can embed its constituents to allow us to $\leq_m^p$-reduce from control problems about its constituents to control problems about *hybrid*.

Theorem 3.3 *Let $k \in \mathbb{N}^+$ and let $\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1}$ be election systems. Let Φ be one of the standard twenty types of (constructive and destructive) control. If for at least one i, $0 \leq i \leq k-1$, $\mathcal{E}_i$ is candidate-anonymous and resistant to Φ, then hybrid$(\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_{k-1})$ is resistant to Φ.*

Corollary 3.4 *hybrid strongly inherits resistance to each of the standard twenty types of control.*

Before we turn to applying this corollary, let us note that Theorem 3.3 and Corollary 3.4 are both, as is this entire paper, within the most natural, most typical model: Votes are input as a list ("nonsuccinct" input) and each vote counts equally ("unweighted" votes). We mention that for each of the other three cases—"succinct, weighted," "succinct, unweighted," and "nonsuccinct, weighted"—Theorem 3.3 and Corollary 3.4 both still hold.

Let us apply Corollary 3.4 to obtain election systems that are broadly resistant to control.

Corollary 3.5 *There exist election systems—for example, hybrid*(plurality, Condorcet)*—that are resistant to all the standard ten types of constructive control.*

To make the same claim for destructive control, a bit more work is needed, since for three of the standard ten types of destructive control no system has been, as far as we know, proven to be resistant. So we first construct an artificial system, $\mathcal{E}_{\text{not-all-one}}$ (defined in the full version), having the missing three resistance properties.

Lemma 3.6 *There exists a candidate-anonymous election system,* $\mathcal{E}_{\text{not-all-one}}$*, that is resistant to (a) destructive control by deleting voters, (b) destructive control by adding voters, and (c) destructive control by partition of voters in the TE model.*

Corollary 3.7 *There exist election systems that are resistant to all ten standard types of destructive control.*

We cannot apply Theorem 3.3 directly to rehybridize the systems of Corollaries 3.5 and 3.7, because *hybrid* itself is not in general candidate-anonymous. However, we can get the same conclusion by directly hybridizing all the constituents underlying Corollaries 3.5 and 3.7.

Theorem 3.8 *There exist election systems that are resistant to all twenty standard types of control.*

The proof simply is to consider *hybrid*(plurality, Condorcet, $\mathcal{E}_{\text{not-all-one}}$).

3.3 Immunity

We now turn to inheritance of immunity. Here, for each of constructive and destructive control, five cases inherit and five cases provably fail to inherit.

Theorem 3.9 *Any candidate-anonymous election system that is immune to constructive control by deleting candidates can never have a unique winner.*

Since "never having a unique winner" is inherited by *hybrid*, Theorem 3.9 implies:

Theorem 3.10 *Immunity to constructive control by deleting candidates is inherited by hybrid.*

By applying a duality result of Hemaspaandra, Hemaspaandra, and Rothe multiple times, we can retarget this to a type of destructive control.

Proposition 3.11 ([HHR05b]) *A voting system is susceptible to constructive control by deleting candidates if and only if it is susceptible to destructive control by adding candidates.*

Corollary 3.12 *Immunity to destructive control by adding candidates is inherited by hybrid.*

hybrid's immunity to all voter-related types of control is immediate.

Theorem 3.13 *Immunity to constructive and destructive control under each of (a) adding voters, (b) deleting voters, (c) partition of voters in model TE, and (d) partition of voters in model TP is inherited (and flexibly so) by hybrid.*

For the ten remaining cases, inheritance does not hold.

3.4 Vulnerability

hybrid strongly inherited resistance, which is precisely what one wants, since that is both the aesthetically pleasing case and broadens resistance to control. However, for vulnerability it is less clear what outcome to root for. Inheritance would be the mathematically more beautiful outcome. But on the other hand, what inheritance would inherit is vulnerability, and vulnerability to control is in general a bad thing—so maybe one should hope for "Not I(nherits)" entries for our table in this column. In fact, our results here are mixed. In particular, we for ten cases prove that inheritance holds unconditionally and for ten cases prove that inheritance holds (though in some cases we have to limit ourselves to election systems with winner/unique winner problems that fall into the polynomial hierarchy) if and only if P = NP.

4 Conclusions

Table 1 summarizes our inheritance results. The main contribution of this paper is the *hybrid* system, the fact that *hybrid* strongly inherits resistance, and the consequence that there is an election system that resists all twenty standard types of electoral control. The authors jointly with P. Faliszewski are currently working to show that some natural election systems may exhibit broad resistance to control.

Acknowledgments: We thank Holger Spakowski, COMSOC '06 referees, and IJCAI '07 referees for helpful comments.

References

[Arr63] K. Arrow. *Social Choice and Individual Values.* John Wiley and Sons, 1951 (revised edition 1963).

[BO91] J. Bartholdi III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[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.

[CLS03] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 201–214. ACM Press, 2003.

[CS02] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 314–319. AAAI Press, 2002.

[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.

[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.

[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.

[EL05] E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In *Proceedings of the 16th International Symposium on Algorithms and Computation*, pages 206–215. Springer-Verlag *Lecture Notes in Computer Science #3827*, December 2005.

[ER91] E. Ephrati and J. Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *Proceedings of the 9th National Conference on Artificial Intelligence*, pages 173–178. AAAI Press, 1991.

[ER93] E. Ephrati and J. Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 423–429, 1993.

[FHH06] 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.

[FKS03] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 301–312. ACM Press, 2003.

[Gib73] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.

[HHR05a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 95–101. AAAI Press, 2005.

[HHR05b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. Technical Report cs.GT/0507027, Computing Research Repository, http://www.acm.org/repository/, July 2005. Revised, May 2006.

[HHR06] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. Technical Report TR-900, Department of Computer Science, University of Rochester, Rochester, NY, June 2006. Revised, August 2006. Conference version to appear in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*.

[PHG00] D. Pennock, E. Horvitz, and C. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 729–734. AAAI Press, 2000.

[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.

[SV00] H. Spakowski and J. Vogel. Θ_2^p-completeness: A classical approach for new results. In *Proceedings of the 20th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 348–360. Springer-Verlag *Lecture Notes in Computer Science #1974*, December 2000.

[Tay05] A. Taylor. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, 2005.

Edith Hemaspaandra
Department of Computer Science, Rochester Institute of Technology
Rochester, NY 14623, USA, `eh` "at" `cs.rit.edu`.

Lane A. Hemaspaandra
Department of Computer Science, University of Rochester
Rochester, NY 14627, USA, `lane` "at" `cs.rochester.edu`.

Jörg Rothe
Institut für Informatik, Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany, `rothe` "at" `cs.uni-duesseldorf.de`.