The Complexity of Online Manipulation of Sequential Elections

Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe

Abstract

Most work on manipulation assumes that all preferences are known to the manipulators. However, in many settings elections are open and sequential, and manipulators may know the already cast votes but may not know the future votes. We introduce a framework, in which manipulators can see the past votes but not the future ones, to model online coalitional manipulation of sequential elections, and we show that in this setting manipulation can be extremely complex even for election systems with simple winner problems. Yet we also show that for some of the most important election systems such manipulation is simple in certain settings. This suggests that when using sequential voting, one should pay great attention to the details of the setting in choosing one's voting rule.

Among the highlights of our classifications are: We show that, depending on the size of the manipulative coalition, the online manipulation problem can be complete for each level of the polynomial hierarchy or even for PSPACE. And we obtain the most dramatic contrast to date between the nonunique-winner and unique-winner models: Online weighted manipulation for plurality is in P in the nonunique-winner model, yet is coNP-hard (constructive case) and NP-hard (destructive case) in the unique-winner model.

1 Introduction

Voting is a widely used method for preference aggregation and decision-making. In particular, *strategic* voting (or *manipulation*) has been studied intensely in social choice theory (starting with the celebrated work of Gibbard [Gib73] and Satterthwaite [Sat75]) and, in the rapidly emerging area of *computational* social choice, also with respect to its algorithmic properties and computational complexity (starting with the seminal work of Bartholdi, Tovey, and Trick [BTT89]; see the recent surveys by Faliszewski et al. [FP10, FHH10, FHHR09]). This computational aspect is particularly important in light of the many applications of voting in computer science, ranging from meta-search heuristics for the internet [DKNS01], to recommender systems [GMHS99] and multiagent systems in artificial intelligence (see the survey by Conitzer [Con10]).

Most of the previous work on manipulation, however, is concerned with voting where the manipulators know the nonmanipulative votes. Far less attention has been paid (see the related work below) to manipulation in the midst of elections that are modeled as dynamic processes.

We introduce a novel framework for online manipulation, where voters vote in sequence and the current manipulator, who knows the previous votes and which voters are still to come but does not know their votes, must decide—right at that moment—what the "best" vote to cast is. So, while other approaches to sequential voting are game-theoretic, stochastic, or axiomatic in nature (again, see the related work), our approach to manipulation of sequential voting is shaped by the area of "online algorithms" [BE98], in the technical sense of a setting in which one (for us, each manipulative voter) is being asked to make a manipulation decision just on the basis of the information one has in one's hands at the moment even though additional information/system evolution may well be happening down the line. In this area, there are different frameworks for evaluation. But the most attractive one, which pervades the area as a general theme, is the idea that one may want to "maxi-min" things—*one may want to take the action that maximizes the goodness of the set of outcomes that one can expect regardless of what happens down the line from one time-wise*. For example, if the current manipulator's preferences are Alice $>$ Ted $>$ Carol $>$ Bob and if she can cast a (perhaps insincere) vote that ensures that Alice or Ted will be a winner no matter what later voters do, and there is no vote she can cast that ensures that Alice will always be a winner, this maxi-min approach would say that that vote is a "best" vote to cast.

It will perhaps be a bit surprising to those familiar with online algorithms and competitive analysis that in our model of online manipulation we will not use a (competitive) *ratio*. The reason is that voting commonly uses an *ordinal* preference model, in which preferences are total orders of the candidates. It would be a severely improper step to jump from that to assumptions about intensity of preferences and utility, e.g., to assuming that everyone likes her *n*th-to-least favorite candidate exactly n times more than she likes her least favorite candidate.

Related Work. Conitzer and Xia [XC10a] (see also the related paper by Desmedt and Elkind [DE10]) define and study the Stackelberg voting game (also quite naturally called, in an earlier paper that mostly looked at two candidates, the roll-call voting game [Slo93]). This basically is an election in which the voters vote in order, *and the preferences are common knowledge—everyone knows everyone else's preferences, everyone knows that everyone knows everyone else's preferences, and so on out to infinity*. Their analysis of this game is fundamentally game-theoretic; with such complete knowledge in a sequential setting, there is precisely one (subgame perfect Nash) equilibrium, which can be computed from the back end forward. Under their work's setting and assumptions, for bounded numbers of manipulators manipulation is in P, but we will show that in our model even with bounded numbers of manipulators manipulation sometimes (unless $\mathrm{P} = \mathrm{NP}$) falls beyond P.

The interesting "dynamic voting" work of Tennenholtz [Ten04] investigates sequential voting, but focuses on axioms and voting rules rather than on coalitions and manipulation. Much heavily Markovian work studies sequential decision-making and/or dynamically varying preferences (see [PP11] and the references therein); our work in contrast is nonprobabilistic and focused on the complexity of coalitional manipulation. Also somewhat related to, but quite different from, our work is the work on possible and necessary winners. The seminal paper on that is due to Konczak and Lang [KL05], and more recent work includes [XC08, BHN09, BBF10, Bet10, BD10, CLM$^+$12, BR12]; the biggest difference is that those are, loosely, one-quantifier settings, but the more dynamic setting of online manipulation involves numbers of quantifiers that can grow with the input size. Another related research line studies multi-issue elections [XC10b, XCL10, XCL11, XLC11]; although there the separate issues may run in sequence, each issue typically is voted on simultaneously and with preferences being common knowledge.

2 Preliminaries

Elections. A *(standard, i.e., simultaneous) election* (C,V) is specified by a set C of candidates and a list V, where we assume that each element in V is a pair (v,p) such that v is a voter name and p is v's vote. How the votes in V are represented depends on the election system used—we assume, as is required by most systems, votes to be total preference orders over C. For example, if $C = \{a,b,c\}$, a vote of the form $c > a > b$ means that this voter (strictly) prefers c to a and a to b.

We introduce election snapshots to capture sequential election scenarios as follows. Let C be a set of candidates and let u be (the name of) a voter. An *election snapshot for C and u* is specified by a triple $V = (V_{<u}, u, V_{u<})$ consisting of all voters in the order they vote, along with, for each voter before u (i.e., those in $V_{<u}$), the vote she cast, and for each voter after u (i.e., those in $V_{u<}$), a bit specifying if she is part of the manipulative coalition (to which u always belongs). That is, $V_{<u} = ((v_1, p_1), (v_2, p_2), \ldots, (v_{i-1}, p_{i-1}))$, where the voters named $v_1, v_2, \ldots, v_{i-1}$ (including perhaps manipulators and nonmanipulators) have already cast their votes (preference order p_j being cast by v_j), and $V_{u<} = ((v_{i+1}, x_{i+1}), (v_{i+2}, x_{i+2}), \ldots, (v_n, x_n))$ lists the names of the voters still to cast their votes, in that order, and where $x_j = 1$ if v_j belongs to the manipulative coalition and $x_j = 0$ otherwise.

Scoring Rules. A *scoring rule* for m candidates is given by a scoring vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ of nonnegative integers such that $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$. For an election (C,V), each candidate $c \in C$ scores α_i points for each vote that ranks c in the ith position. Let $score(c)$ be the total score of $c \in C$. All candidates scoring the most points are winners of (C,V). Some of the most popular voting systems are *k-approval* (especially *plurality*, aka 1-approval) and *k-veto* (especially *veto*, aka 1-veto). Their m-candidate, $m \geq k$, versions are defined by the scoring vectors $(\underbrace{1,\ldots,1}_{k},\underbrace{0,\ldots,0}_{m-k})$ and $(\underbrace{1,\ldots,1}_{m-k},\underbrace{0,\ldots,0}_{k})$. When m is not fixed, we omit the phrase “m-candidate.”

Manipulation. The *(standard) weighted coalitional manipulation problem* [CSL07], $\mathscr{E}$-Weighted-Coalitional-Manipulation (abbreviated by $\mathscr{E}$-WCM), for any election system $\mathscr{E}$ is defined as follows: Given a candidate set C, a list S of nonmanipulative voters each having a nonnegative integer weight, a list T of the nonnegative integer weights of the manipulative voters (whose preferences over C are unspecified), with $S \cap T = \emptyset$, and a distinguished candidate $c \in C$, can the manipulative votes T be set such that c is a (or the) $\mathscr{E}$ winner of $(C, S \cup T)$?

Asking whether c can be made “a winner” is called the nonunique-winner model and is the model of all notions in this paper unless mentioned otherwise. If one asks whether c can be made a “one and only winner,” that is called the unique-winner model. We also use the *unweighted* variant, where each vote has unit weight, and write $\mathscr{E}$-UCM as a shorthand. Note that $\mathscr{E}$-UCM with a *single* manipulator (i.e., $\|T\| = 1$ in the problem instance) is the manipulation problem originally studied in [BTT89, BO91]. Conitzer, Sandholm, and Lang [CSL07] also introduced the *destructive* variants of these manipulation problems, where the goal is not to make c win but to ensure that c is not a winner, and we denote the corresponding problems by $\mathscr{E}$-DWCM and $\mathscr{E}$-DUCM. Finally, we write $\mathscr{E}$-WC$_{\neq\emptyset}$M, $\mathscr{E}$-UC$_{\neq\emptyset}$M, $\mathscr{E}$-DWC$_{\neq\emptyset}$M, and $\mathscr{E}$-DUC$_{\neq\emptyset}$M to indicate that the problem instances are required to have a nonempty coalition of manipulators.

Complexity-Theoretic Background. We assume the reader is familiar with basic complexity-theoretic notions such as the complexity classes P and NP, the class FP of polynomial-time computable functions, polynomial-time many-one reducibility ($\leq_m^p$), and hardness and completeness with respect to $\leq_m^p$ for a complexity class.

Meyer and Stockmeyer [MS72] and Stockmeyer [Sto76] introduced and studied the polynomial hierarchy, $\mathrm{PH} = \bigcup_{k\geq 0} \Sigma_k^p$, whose levels are inductively defined by $\Sigma_0^p = \mathrm{P}$ and $\Sigma_{k+1}^p = \mathrm{NP}^{\Sigma_k^p}$, and their co-classes, $\Pi_k^p = \mathrm{co}\Sigma_k^p$ for $k \geq 0$. They also characterized these levels by polynomially length-bounded alternating existential and universal quantifiers. P^{NP} is the class of problems solvable in deterministic polynomial time with access to an NP oracle, and $\mathrm{P}^{\mathrm{NP}[1]}$ is the restriction of P^{NP} where only one oracle query is allowed. Note that $\mathrm{P} \subseteq \mathrm{NP} \cap \mathrm{coNP} \subseteq \mathrm{NP} \cup \mathrm{coNP} \subseteq \mathrm{P}^{\mathrm{NP}[1]} \subseteq \mathrm{P}^{\mathrm{NP}} \subseteq \Sigma_2^p \cap \Pi_2^p \subseteq \Sigma_2^p \cup \Pi_2^p \subseteq \mathrm{PH} \subseteq \mathrm{PSPACE}$, where PSPACE is the class of problems solvable in polynomial space. The *quantified boolean formula problem*, QBF, is a standard PSPACE-complete problem. Define QBF_k ($\widetilde{\mathrm{QBF}}_k$) to be the restriction of QBF with at most k quantifiers that start with $\exists$ ($\forall$) and then alternate between $\exists$ and $\forall$, and we assume that each $\exists$ and $\forall$ quantifies over a set of boolean variables. For each $k \geq 1$, QBF_k is Σ_k^p-complete and $\widetilde{\mathrm{QBF}}_k$ is Π_k^p-complete.

Proofs omitted due to space limitations can be found in the technical report version [HHR12a].

3 Our Model of Online Manipulation

The core of our model of online manipulation in sequential voting is what we call the *magnifying-glass moment*, namely, the moment at which a manipulator u is the one who is going to vote, is aware of what has happened so far in the election (and which voters are still to come, but in general not knowing what they want, except in the case of voters, if any, who are coalitionally linked to u). In this moment, u seeks to "figure out" what the "best" vote to cast is. We will call the information available in such a moment an *online manipulation setting* (*OMS*, for short) and define it formally as a tuple (C,u,V,σ,d), where C is a set of candidates; u is a distinguished voter; $V=(V_{<u},u,V_{u<})$ is an election snapshot for C and u; σ is the preference order of the manipulative coalition to which u belongs; and $d \in C$ is a distinguished candidate. Given an election system $\mathscr{E}$, define the problem online-$\mathscr{E}$-Unweighted-Coalitional-Manipulation (abbreviated by online-$\mathscr{E}$-UCM), as follows: Given an OMS (C,u,V,σ,d) as described above, does there exist some vote that u can cast (assuming support from the manipulators coming after u) such that no matter what votes are cast by the nonmanipulators coming after u, there exists some $c \in C$ such that $c \geq_\sigma d$ and c is an $\mathscr{E}$ winner of the election? By "support from the manipulators coming after u" we mean that u's coalition partners coming after u, when they get to vote, will use their then-in-hand knowledge of all votes up to then to help u reach her goal: By a joint effort u's coalition can ensure that the $\mathscr{E}$ winner set will always include a candidate liked by the coalition as much as or more than d, even when the nonmanipulators take their strongest action so as to prevent this. Note that this candidate, c in the problem description, may be different based on the nonmanipulators' actions. (Nonsequential manipulation problems usually focus on whether a single candidate can be made to win, but in our setting, this "that person or better" focus is more natural.) For the case of weighted manipulation, each voter also comes with a nonnegative integer weight. We denote this problem by online-$\mathscr{E}$-WCM.

We write online-$\mathscr{E}$-UCM$[k]$ in the unweighted case and online-$\mathscr{E}$-WCM$[k]$ in the weighted case to denote the problem when the number of manipulators from u onward is restricted to be at most k.

Our corresponding destructive problems are denoted by online-$\mathscr{E}$-DUCM, online-$\mathscr{E}$-DWCM, online-$\mathscr{E}$-DUCM$[k]$, and online-$\mathscr{E}$-DWCM$[k]$. In online-$\mathscr{E}$-DUCM we ask whether the given current manipulator u (assuming support from the manipulators after her) can cast a vote such that no matter what votes are cast by the nonmanipulators after u, no $c \in C$ with $d \geq_\sigma c$ is an $\mathscr{E}$ winner of the election, i.e., u's coalition can ensure that the $\mathscr{E}$ winner set never includes d or any even more hated candidate. The other three problems are defined analogously.

Note that online-$\mathscr{E}$-UCM generalizes the original unweighted manipulation problem with a single manipulator as introduced by Bartholdi, Tovey, and Trick [BTT89]. Indeed, their manipulation problem in effect is the special case of online-$\mathscr{E}$-UCM when restricted to instances where there is just one manipulator, she is the last voter to cast a vote, and d is the coalition's most preferred candidate. Similarly, online-$\mathscr{E}$-WCM generalizes the (standard) coalitional weighted manipulation problem (for nonempty coalitions of manipulators). Indeed, that traditional manipulation problem is the special case of online-$\mathscr{E}$-WCM, restricted to instances where only manipulators come after u and d is the coalition's most preferred candidate. If we take an analogous approach except with d restricted now to being the most hated candidate of the coalition, we generalize the corresponding notions for the destructive cases. We summarize these observations as follows.

Proposition 1 *For each election system $\mathscr{E}$, it holds that (1)* $\mathscr{E}$-UC$_{\neq\emptyset}$M $\leq_m^p$ online-$\mathscr{E}$-UCM, *(2)* $\mathscr{E}$-WC$_{\neq\emptyset}$M $\leq_m^p$ online-$\mathscr{E}$-WCM, *(3)* $\mathscr{E}$-DUC$_{\neq\emptyset}$M $\leq_m^p$ online-$\mathscr{E}$-DUCM, *and (4)* $\mathscr{E}$-DWC$_{\neq\emptyset}$M $\leq_m^p$ online-$\mathscr{E}$-DWCM.

Corollary 2 below follows immediately from the above proposition.

Corollary 2 *(1) For each election system $\mathscr{E}$ such that the (unweighted) winner problem is solvable in polynomial time, it holds that* $\mathscr{E}$-UCM $\leq_m^p$ online-$\mathscr{E}$-UCM. *(2) For each election system $\mathscr{E}$ such that the weighted winner problem is solvable in polynomial time, it holds that* $\mathscr{E}$-WCM $\leq_m^p$ online-$\mathscr{E}$-WCM. *(3) For each election system $\mathscr{E}$ such that the winner problem is solvable in polynomial time, it holds that* $\mathscr{E}$-DUCM $\leq_m^p$ online-$\mathscr{E}$-DUCM. *(4) For each election system $\mathscr{E}$ such that the weighted winner problem is solvable in polynomial time, it holds that* $\mathscr{E}$-DWCM $\leq_m^p$ online-$\mathscr{E}$-DWCM.

We said above that, by default, we will use the *nonunique-winner model* and all the above problems are defined in this model. However, we will also have some results in the *unique-winner model*, which will, here, sharply contrast with the corresponding results in the nonunique-winner model. To indicate that a problem, such as online-$\mathscr{E}$-UCM, is in the unique-winner model, we write online-$\mathscr{E}$-UCM$_{\mathrm{UW}}$ and ask whether the current manipulator u (assuming support from the manipulators coming after her) can ensure that there exists some $c \in C$ such that $c \geq_\sigma d$ and c is *the unique* $\mathscr{E}$ *winner* of the election.

4 General Results

Theorem 3 *(1) For each election system $\mathscr{E}$ whose weighted winner problem can be solved in polynomial time,*[1] *the problem* online-$\mathscr{E}$-WCM *is in* PSPACE*. (2) For each election system $\mathscr{E}$ whose winner problem can be solved in polynomial time, the problem* online-$\mathscr{E}$-UCM *is in* PSPACE*. (3) There exists an election system $\mathscr{E}$ with a polynomial-time winner problem such that the problem* online-$\mathscr{E}$-UCM *is* PSPACE*-complete. (4) There exists an election system $\mathscr{E}$ with a polynomial-time weighted winner problem such that the problem* online-$\mathscr{E}$-WCM *is* PSPACE*-complete.*

PROOF. The proof of the first statement (which is analogous to the proof of the first statement in Theorem 4) follows from the easy fact that online-$\mathscr{E}$-WCM can be solved by an alternating Turing machine in polynomial time, and thus, due to the characterization of Chandra, Kozen, and Stockmeyer [CKS81], by a deterministic Turing machine in polynomial space. The proof of the second case is analogous.

We construct an election system $\mathscr{E}$ establishing the third statement. Let (C,u,V,σ,d) be a given input. $\mathscr{E}$ will look at the lexicographically least candidate name in C. Let c represent that name string in some fixed, natural encoding. $\mathscr{E}$ will check if c represents a *tiered* boolean formula, by which we mean one whose variable names are all of the form $x_{i,j}$ (which really means a direct encoding of a string, such as "$x_{4,9}$"); the i, j fields must all be positive integers. If c does not represent such a tiered formula, everyone loses on that input. Otherwise (i.e., if c represents a tiered formula), let *width* be the maximum j occurring as the second subscript in any variable name ($x_{i,j}$) in c, and let *blocks* be the maximum i occurring as the first subscript in any variable name in c. If there are fewer than *blocks* voters in V, everyone loses. Otherwise, if there are fewer than $1+2\cdot \mathit{width}$ candidates in C, everyone loses (this is so that each vote will involve enough candidates that it can be used to set all the variables in one block). Otherwise, if there exists some i, $1 \leq i \leq \mathit{blocks}$, such that for no j does the variable $x_{i,j}$ occur in c, then everyone loses. Otherwise, order the voters from the lexicographically least to the lexicographically greatest voter name. If distinct voters are allowed to have the same name string (e.g., John Smith), we break ties by sorting according to the associated preference orders within each group of tied voters (second-order ties are no problem, as those votes are identical, so any order will have the same effect). Now, the first voter in this order will assign truth values to all variables $x_{1,*}$, the second voter in this order will assign truth values to all variables $x_{2,*}$, and so on up to the *blocks*th voter, who will assign truth values to all variables $x_{\mathit{blocks},*}$.

How do we get those assignments from these votes? Consider a vote whose total order over C is σ' (and recall that $\|C\| \geq 1+2\cdot \mathit{width}$). Remove c from σ', yielding σ''. Let $c_1 <_{\sigma''} c_2 <_{\sigma''} \cdots <_{\sigma''} c_{2\cdot \mathit{width}}$ be the $2\cdot \mathit{width}$ least preferred candidates in σ''. We build a vector in $\{0,1\}^{\mathit{width}}$ as follows: The ℓth bit of the vector is 0 if the string that names $c_{1+2(\ell-1)}$ is lexicographically less than the string that names $c_{2\ell}$, and this bit is 1 otherwise.

Let b_i denote the vector thus built from the ith vote (in the above ordering), $1 \leq i \leq \mathit{blocks}$. Now, for each variable $x_{i,j}$ occurring in c, assign to it the value of the jth bit of b_i, where 0 represents *false*

[1]We mention in passing here, and henceforward we will not explicitly mention it in the analogous cases, that the claim clearly remains true even when "polynomial time" is replaced by the larger class "polynomial space."

and 1 represents *true*. We have now assigned all variables of c, so c evaluates to either *true* or *false*. If c evaluates to *true*, everyone wins, otherwise everyone loses. This completes the specification of the election system $\mathscr{E}$. $\mathscr{E}$ has a polynomial-time winner problem, as any boolean formula, given an assignment to all its variables, can easily be evaluated in polynomial time.

To show PSPACE-hardness, we $\leq_m^p$-reduce the PSPACE-complete problem QBF to the problem online-$\mathscr{E}$-UCM. Let y be an instance of QBF. We transform y into an instance of the form

$$(\exists x_{1,1}, x_{1,2}, \ldots, x_{1,k_1})\,(\forall x_{2,1}, x_{2,2}, \ldots, x_{2,k_2}) \cdots (Q_\ell x_{\ell,1}, x_{\ell,2}, \ldots, x_{\ell,k_\ell})$$
$$[\Phi(x_{1,1}, x_{1,2}, \ldots, x_{1,k_1}, x_{2,1}, x_{2,2}, \ldots, x_{2,k_2}, \ldots, x_{\ell,1}, x_{\ell,2}, \ldots, x_{\ell,k_\ell})]$$

in polynomial time, where $Q_\ell = \exists$ if ℓ is odd and $Q_\ell = \forall$ if ℓ is even, the $x_{i,j}$ are boolean variables, Φ is a boolean formula, and for each i, $1 \leq i \leq \ell$, Φ contains at least one variable of the form $x_{i,*}$. This quantified boolean formula is $\leq_m^p$-reduced to an instance (C, u, V, σ, c) of online-$\mathscr{E}$-UCM as follows:

1. C contains a candidate whose name, c, encodes Φ, and in addition C contains $2 \cdot \max(k_1, \ldots, k_\ell)$ other candidates, all with names lexicographically greater than c—for specificity, let us say their names are the $2 \cdot \max(k_1, \ldots, k_\ell)$ strings that immediately follow c in lexicographic order.

2. V contains ℓ voters, $1, 2, \ldots, \ell$, who vote in that order, where $u = 1$ is the distinguished voter and all odd voters belong to u's manipulative coalition and all even voters do not. The voter names will be lexicographically ordered by their number, 1 is least and ℓ is greatest.

3. The manipulators' preference order σ is to like candidates in the opposite of their lexicographic order. In particular, c is the coalition's most preferred candidate.

This is a polynomial-time reduction. It follows immediately from this construction and the definition of $\mathscr{E}$ that y is in QBF if and only if (C, u, V, σ, c) is in online-$\mathscr{E}$-UCM.

To prove the last statement, simply let $\mathscr{E}$ be the election system that ignores the weights of the voters and then works exactly as the previous election system. ❑

The following theorem shows that for bounded numbers of manipulators the complexity crawls up the polynomial hierarchy. The theorem's proof is based on the proof given above, except we need to use the alternating quantifier characterization due to Meyer and Stockmeyer [MS72] and Stockmeyer [Sto76] for the upper bound and to reduce from the Σ_{2k}^p-complete problem QBF$_{2k}$ rather than from QBF for the lower bound.

Theorem 4 *Fix any $k \geq 1$. (1) For each election system $\mathscr{E}$ whose weighted winner problem can be solved in polynomial time, the problem* online-$\mathscr{E}$-WCM$[k]$ *is in Σ_{2k}^p. (2) For each election system $\mathscr{E}$ whose winner problem can be solved in polynomial time, the problem* online-$\mathscr{E}$-UCM$[k]$ *is in Σ_{2k}^p. (3) There exists an election system $\mathscr{E}$ with a polynomial-time winner problem such that the problem* online-$\mathscr{E}$-UCM$[k]$ *is Σ_{2k}^p-complete. (4) There exists an election system $\mathscr{E}$ with a polynomial-time weighted winner problem such that the problem* online-$\mathscr{E}$-WCM$[k]$ *is Σ_{2k}^p-complete.*

Note that the (constructive) online manipulation problems considered in Theorems 3 and 4 are about ensuring that the winner set always contains some candidate in the σ segment stretching from d up to the top-choice. Now consider "pinpoint" variants of these problems, where we ask whether the distinguished candidate d herself can be guaranteed to be a winner (for nonsequential manipulation, that version indeed is the one commonly studied). Denote the *pinpoint* variant of, e.g., online-$\mathscr{E}$-UCM$[k]$ by pinpoint-online-$\mathscr{E}$-UCM$[k]$. Since our hardness proofs in Theorems 3 and 4 make all or no one a winner (and as the upper bounds in these theorems also can be seen to hold for the pinpoint variants), they establish the corresponding completeness results also for the pinpoint cases. We thus have completeness results for PSPACE and Σ_{2k}^p for each $k \geq 1$. What about the classes Σ_{2k-1}^p and Π_k^p, for each $k \geq 1$? We can get completeness results for all these classes by defining appropriate variants of online manipulation problems. Let OMP be any of the online manipulation problems considered earlier, including the pinpoint variants mentioned above. Define freeform-OMP to be just as OMP, except we no longer require the distinguished voter u to be part of the manipulative coalition—u can be in or can be out, and the input must specify, for u and all voters after u, which ones are the members of the coalition. The question of freeform-OMP is whether it is true that for all actions of the nonmanipulators at or after u (for specificity as to this problem: if u is a nonmanipulator, it will in the input come with a preference order) there will be actions (each taken with full information on cast-before-them votes) of the manipulative coalition members such that their goal of making some candidate c with $c \geq_\sigma d$ (or exactly d, in the pinpoint versions) a winner is achieved. Then, whenever Theorem 4 establishes a Σ_{2k}^p or Σ_{2k}^p-completeness result for OMP, we obtain a Π_{2k+1}^p or Π_{2k+1}^p-completeness result for freeform-OMP and for $k = 0$ manipulators we obtain $\Pi_1^p = \mathrm{coNP}$ or coNP-completeness results. Similarly, the PSPACE and PSPACE-completeness results for OMP we established in Theorem 3 also can be shown true for freeform-OMP.

On the other hand, if we define a variant of OMP by requiring the final voter to always be a manipulator, the PSPACE and PSPACE-completeness results for OMP from Theorem 3 remain true for this variant; the Σ_{2k}^p and Σ_{2k}^p-completeness results for OMP from Theorem 4 change to Σ_{2k-1}^p and Σ_{2k-1}^p-completeness results for this variant; and the above Π_{2k+1}^p and Π_{2k+1}^p-completeness results for freeform-OMP change to Π_{2k}^p and Π_{2k}^p-completeness results for this variant, $k \geq 1$.

Finally, as an open direction (and related conjecture), we define for each of the previously considered variants of online manipulation problems a *full profile* version. For example, for a given election system $\mathscr{E}$, fullprofile-online-$\mathscr{E}$-UCM$[k]$ is the function problem that, given an OMS *without* any distinguished candidate, (C, u, V, σ), returns a length $\|C\|$ bit-vector that for each candidate $d \in C$ says if the answer to "$(C, u, V, \sigma, d) \in$ online-$\mathscr{E}$-UCM$[k]$?" is "yes" (1) or "no" (0). The function problem fullprofile-pinpoint-online-$\mathscr{E}$-UCM$[k]$ is defined analogously, except regarding pinpoint-online-$\mathscr{E}$-UCM$[k]$.

It is not hard to prove, as a corollary to Theorem 4, that:

Theorem 5 *For each election system $\mathscr{E}$ whose winner problem can be solved in polynomial time, (1)* fullprofile-online-$\mathscr{E}$-UCM$[k]$ *is in* $\mathrm{FP}^{\Sigma_{2k}^p[\mathscr{O}(\log n)]}$*, the class of functions computable in polynomial time given Turing access to a* Σ_{2k}^p *oracle with* $\mathscr{O}(\log n)$ *queries allowed on inputs of size n, and (2)* fullprofile-pinpoint-online-$\mathscr{E}$-UCM$[k]$ *is in* $\mathrm{FP}_{\mathrm{tt}}^{\Sigma_{2k}^p}$*, the class of functions computable in polynomial time given truth-table access to a* Σ_{2k}^p *oracle.*

We conjecture that both problems are complete for the corresponding class under metric reductions [Kre88], for suitably defined election systems with polynomial-time winner problems.

If the full profile version of an online manipulation problem can be computed efficiently, we clearly can also easily solve each of the decision problems involved by looking at the corresponding bit of the length $\|C\|$ bit-vector. Conversely, if there is an efficient algorithm for an online manipulation decision problem, we can easily solve its full profile version by running this algorithm for each candidate in turn. Thus, we will state our later results only for online manipulation decision problem.

Proposition 6 *Let* OMP *be any of the online manipulation decision problems defined above. Then* fullprofile-OMP *is in* FP *if and only if* OMP *is in* P.

5 Results for Specific Natural Voting Systems

The results of the previous section show that, simply put, even for election systems with polynomial-time winner problems, online manipulation can be tremendously difficult. But what about *natural* election systems? We will now take a closer look at important natural systems. We will show that online manipulation can be easy for them, depending on which particular problem is considered, and we will also see that the constructive and destructive cases can differ sharply from each other and that it really matters whether we are in the nonunique-winner model or the unique-winner model.

Theorem 7 *(1)* online-plurality-WCM *(and thus also* online-plurality-UCM*) is in* P. *(2)* online-plurality-DWCM *(and thus also* online-plurality-DUCM*) is in* P.

Theorem 7 refers to problems in the nonunique-winner model. By contrast, we now show that online manipulation for weighted plurality voting in the *unique-winner* model is coNP-hard in the *constructive* case and is NP-hard in the *destructive* case. This is perhaps the most dramatic, broad contrast yet between the nonunique-winner model and the unique-winner model, and is the first such contrast involving plurality. The key other NP-hardness versus P result for the nonunique-winner model versus the unique-winner model is due to Faliszewski, Hemaspaandra, and Schnoor [FHS08], but holds only for (standard) weighted manipulation for Copeland$^{\alpha}$ elections ($0 < \alpha < 1$) with exactly three candidates; for fewer than three both cases there are in P and for more than three both are NP-complete. In contrast, the P results of Theorem 7 hold for all numbers of candidates, and the NP-hardness and coNP-hardness results of Theorem 8 hold whenever there are at least two candidates.

Theorem 8 *(1) The problem* online-plurality-DWCM$_{\text{UW}}$ *is* NP*-hard, even when restricted to only two candidates (and this also holds when restricted to three, four, ... candidates). (2) The problem* online-plurality-WCM$_{\text{UW}}$ *is* coNP*-hard, even when restricted to only two candidates (and this also holds when restricted to three, four, ... candidates).*

Proof. For the first statement, we prove NP-hardness of online-plurality-DWCM$_{\text{UW}}$ by a reduction from the NP-complete problem Partition: Given a nonempty sequence $(w_1, w_2, \ldots, w_z)$ of positive integers such that $\sum_{i=1}^{z} w_i = 2W$ for some positive integer W, does there exist a set $I \subseteq \{1, 2, \ldots, z\}$ such that $\sum_{i \in I} w_i = W$? Let $m \geq 2$. Given an instance $(w_1, w_2, \ldots, w_z)$ of Partition, construct an instance $(\{c_1, \ldots, c_m\}, u_1, V, c_1 > c_2 > \cdots > c_m, c_1)$ of online-plurality-DWCM$_{\text{UW}}$ such that V contains $m+z-2$ voters $v_1, \ldots, v_{m-2}, u_1, \ldots, u_z$ who vote in that order. For $1 \leq i \leq m-2$, v_i votes for c_i and has weight $(m-1)W - i$, and for $1 \leq i \leq z$, u_i is a manipulator of weight $(m-1)w_i$. If $(w_1, w_2, \ldots, w_z)$ is a yes-instance of Partition, the manipulators can give $(m-1)W$ points to both c_{m-1} and c_m, and zero points to the other candidates. So c_{m-1} and c_m are tied for the most points and there is no unique winner. On the other hand, the only way to avoid having a unique winner in our online-plurality-DWCM$_{\text{UW}}$ instance is if there is a tie for the most points. The only candidates that can tie are c_{m-1} and c_m, since all other pairs of candidates have different scores modulo $m-1$. It is easy to see that c_{m-1} and c_m tie for the most points only if they both get exactly $(m-1)W$ points. It follows that $(w_1, w_2, \ldots, w_z)$ is a yes-instance of Partition.

For the second part, we adapt the above construction to yield a reduction from Partition to the complement of online-plurality-WCM$_{\text{UW}}$. Given an instance $(w_1, w_2, \ldots, w_z)$ of Partition, construct an instance $(\{c_1, \ldots, c_m\}, \widehat{u}, V, c_1 > c_2 > \cdots > c_m, c_m)$ of online-plurality-WCM$_{\text{UW}}$ such that V contains $m+z-1$ voters $v_1, \ldots, v_{m-2}, \widehat{u}, u_1, \ldots, u_z$ who vote in that order. For $1 \leq i \leq m-2$, v_i has the same vote and the same weight as above, $\widehat{u}$ is a manipulator of weight 0, and for $1 \leq i \leq z$, u_i has the same weight as above, but in contrast to the case above, u_i is now a nonmanipulator. By the same argument as above, it follows that $(w_1, w_2, \ldots, w_z)$ is a yes-instance of Partition if and only if the nonmanipulators can ensure that there is no unique winner, which in turn is true if and only if the manipulator can not ensure that there is a unique winner. ❑

Theorem 9 *For each scoring rule* $\alpha = (\alpha_1, \ldots, \alpha_m)$, online-$\alpha$-WCM *is in* P *if* $\alpha_2 = \alpha_m$ *and is* NP*-hard otherwise.*

Theorem 10 *For each k,* online-k-approval-UCM *and* online-k-veto-UCM *are in* P.

PROOF. Consider 1-veto. Given an online-1-veto-UCM instance (C,u,V,σ,d), the best strategy for the manipulators from u onward (let n_1 denote how many of these there are) is to minimize $\max_{c<_\sigma d} score(c)$. Let n_0 denote how many nonmanipulators come after u. We claim that (C,u,V,σ,d) is a yes-instance if and only if d is ranked last in σ or there exists a threshold t such that (1) $\sum_{c<_\sigma d}(maxscore(c) \ominus t) \leq n_1$ (so those manipulators can ensure that all candidates ranked $<_\sigma$ d score at most t points), where "$\ominus$" denotes proper subtraction ($x \ominus y = \max(x-y,0)$) and $maxscore(c)$ is c's score when none of the voters from u onward veto c, and (2) $\sum_{c\geq_\sigma d}(maxscore(c) \ominus (t-1)) > n_0$ (so those nonmanipulators cannot prevent that some candidate ranked $\geq_\sigma$ d scores at least t points).

For 1-veto under the above approach, in each situation where the remaining manipulators can force success against all actions of the remaining nonmanipulators, u (right then as she moves) can set her *and all future manipulators' actions* so as to force success regardless of the actions of the remaining nonmanipulators. For k-approval and k-veto, $k \geq 2$, that approach provably cannot work (as will be explained right after this proof); rather, we sometimes need later manipulators' actions to be shaped by intervening nonmanipulators' actions. Still, the following P-time algorithm, which works for all k, tells whether success can be forced. As a thought experiment, for each voter v from u onwards in sequence do this: Order the candidates in $\{c \mid c \geq_\sigma d\}$ from most to least current approvals, breaking ties arbitrarily, and postpend the remaining candidates ordered from least to most current approvals. Let ℓ be k for k-approval and $\|C\|-k$ for k-veto. Cast the voter's ℓ approvals for the first ℓ candidates in this order if v is a manipulator, and otherwise for the last ℓ candidates in this order. Success can be forced against perfect play if and only if this P-time process leads to success. ❑

In the above proof we said that the approach for 1-veto (in which the current manipulator can set her and all future manipulators' actions so as to force success independent of the actions of intervening future nonmanipulators) provably cannot work for k-approval and k-veto, $k \geq 2$. Why not? Consider an OMS (C,u,V,σ,d) with candidate set $C = \{c_1,c_2,\ldots,c_{2k}\}$, σ being given by $c_1 >_\sigma c_2 >_\sigma \cdots >_\sigma c_{2k}$, and $d = c_1$. So, u's coalition wants to enforce that c_1 is a winner. Suppose that v_1 has already cast her vote, now it's $v_2 = u$'s turn, and the order of the future voters is $v_3,v_4,\ldots,v_{2j}$, where all v_{2i}, $2 \leq i \leq j$, belong to u's coalition, and all v_{2i-1} do not. Suppose that v_1 was approving of the k candidates in $C_1 \subseteq \{c_2,c_3,\ldots,c_{2k}\}$, $\|C_1\| = k$. Then u must approve of the k candidates in $\overline{C_1}$, to ensure that c_1 draws level with the candidates in C_1 and none of these candidates can gain another point. Next, suppose that nonmanipulator v_3 approves of the k candidates in $C_3 \subseteq \{c_2,c_3,\ldots,c_{2k}\}$, $\|C_3\| = k$. Then v_4, the next manipulator, must approve of all candidates in $\overline{C_3}$, to ensure that c_1 draws level with the candidates in C_3 and none of these candidates can gain another point. This process is repeated until the last nonmanipulator, v_{2j-1}, approves of the candidates in $C_{2j-1} \subseteq \{c_2,c_3,\ldots,c_{2k}\}$, $\|C_{2j-1}\| = k$, and v_{2j}, the final manipulator, is forced to counter this by approving of all candidates in $\overline{C_{2j-1}}$, to ensure that c_1 is a winner. This shows that there can be arbitrarily long chains such that the action of each manipulator after u depends on the action of the preceding intervening nonmanipulator.

We now turn to online weighted manipulation for veto when restricted to three candidates. We denote this restriction of online-veto-WCM by online-veto$_{|3}$-WCM.

Theorem 11 online-veto$_{|3}$-WCM *is* $\mathrm{P}^{\mathrm{NP}[1]}$*-complete.*

Dropping the restriction to three candidates, we obtain the following result, which places this problem far below the general PSPACE bound from earlier in this paper. Immediately from Theorems 10 and 12, we have that the full profile variants of online-k-veto-UCM and online-k-approval-UCM are in FP and that fullprofile-online-veto-WCM is in $\mathrm{FP}^{\mathrm{NP}}$.

Theorem 12 online-veto-WCM *is in* P^{NP}.

6 Uncertainty About the Order of Future Voters

So far, we have been dealing with cases where the order of future voters was fixed and known. But what happens if the order of future voters itself is unknown? Even here, we can make claims. To model this most naturally, our "magnifying-glass moment" will focus not on one manipulator u, but will focus at a moment in time when some voters are still to come (as before, we know who they are and which are manipulators; as before, we have a preference order σ, and know what votes have been cast so far, and have a distinguished candidate d). And the question our problem is asking is: Is it the case that our manipulative coalition can ensure that the winner set will always include d or someone liked more than d with respect to σ (i.e., the winner set will have nonempty intersection with $\{c \in C \mid c \geq_\sigma d\}$), *regardless of what order the remaining voters vote in*. We will call this problem the *schedule-robust online manipulation problem*, and will denote it by SR-online-$\mathscr{E}$-UCM. (We will add a "[1,1]" suffix for the restriction of this problem to instances when at most one manipulator and at most one nonmanipulator have not yet voted.) One might think that this problem captures both a Σ_2^p and a Π_2^p issue, and so would be hard for both classes. However, the requirement of schedule robustness tames the problem (basically what underpins that is simply that exists-forall-predicate implies forall-exists-predicate), bringing it into Σ_2^p. Further, we can prove, by explicit construction of such a system, that for some simple election systems this problem is complete for Σ_2^p.

Theorem 13 *(1) For each election system $\mathscr{E}$ whose winner problem is in* P, SR-online-$\mathscr{E}$-UCM *is in Σ_2^p. (2) There exists an election system $\mathscr{E}$, whose winner problem is in* P*, such that* SR-online-$\mathscr{E}$-UCM *(indeed, even* SR-online-$\mathscr{E}$-UCM$[1,1]$*) is Σ_2^p-complete.*

7 Conclusions and Open Questions

We introduced a novel framework for online manipulation in sequential voting, and showed that manipulation there can be tremendously complex even for systems with simple winner problems. We also showed that among the most important election systems, some have efficient online manipulation algorithms but others (unless P = NP) do not. It will be important to, complementing our work, conduct typical-case complexity studies. Also, we have extended the scope of our investigation by studying online control [HHR12c, HHR12b] and will do so by studying online bribery in appropriate models.

Acknowledgments: We thank the reviewers for helpful comments. This work was supported in part by NSF grants CCF-0915792, CCF-1101452, and CCF-1101479, ARC grant DP110101792, DFG grant RO-1202/15-1, an SFF grant of HHU Düsseldorf, Friedrich Wilhelm Bessel Research Awards to Edith Hemaspaandra and Lane A. Hemaspaandra from the Alexander von Humboldt Foundation, and a DAAD grant for a PPP project in the PROCOPE program.

References

[BBF10] Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible winner determination. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 697–702. AAAI Press, July 2010.

[BD10] N. Betzler and B. Dorn. Towards a dichotomy of finding possible winners in elections based on scoring rules. *Journal of Computer and System Sciences*, 76(8):812–836, 2010.

[BE98] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[Bet10] N. Betzler. On problem kernels for possible winner determination under the k-approval protocol. In *Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science*, pages 114–125. Springer-Verlag *Lecture Notes in Computer Science #6281*, August 2010.

[BHN09] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 53–58. AAAI Press, July 2009.

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

[BR12] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. *Information Processing Letters*, 112(5):186–190, 2012.

[BTT89] J. Bartholdi, III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[CKS81] A. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *Journal of the ACM*, 26(1), 1981.

[CLM$^+$12] Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. *Mathematical Social Sciences*, 64(1):74–88, 2012.

[Con10] V. Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010.

[CSL07] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[DE10] Y. Desmedt and E. Elkind. Equilibria of plurality voting with abstentions. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 347–356. ACM Press, June 2010.

[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, March 2001.

[FHH10] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[FHHR09] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, pages 375–406. Springer, 2009.

[FHS08] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 983–990. International Foundation for Autonomous Agents and Multiagent Systems, May 2008.

[FP10] P. Faliszewski and A. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

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

[GMHS99] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: The anatomy of recommender systems. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 434–435. ACM Press, 1999.

[HHR12a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The complexity of online manipulation of sequential elections. Technical Report arXiv:1202.6655 [cs.GT], Computing Research Repository, arXiv.org/corr/, February 2012. Revised May, 2012.

[HHR12b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Controlling candidate-sequential elections. In *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, August 2012. Short paper. To appear.

[HHR12c] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Online voter control in sequential elections. In *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, August 2012. To appear.

[KL05] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, July/August 2005.

[Kre88] M. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.

[MS72] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129. IEEE Press, October 1972.

[PP11] D. Parkes and A. Procaccia. Dynamic social choice: Foundations and algorithms. Working paper, May 2011.

[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.

[Slo93] B. Sloth. The theory of voting and equilibria in noncooperative games. *Games and Economic Behavior*, 5(1):152–169, 1993.

[Sto76] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[Ten04] M. Tennenholtz. Transitive voting. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 230–231. ACM Press, July 2004.

[XC08] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 196–201. AAAI Press, July 2008.

[XC10a] L. Xia and V. Conitzer. Stackelberg voting games: Computational aspects and paradoxes. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 697–702. AAAI Press, July 2010.

[XC10b] L. Xia and V. Conitzer. Strategy-proof voting rules over multi-issue domains with restricted preferences. In *Proceedings of the 6th International Workshop On Internet And Network Economics*, pages 402–414. Springer-Verlag *Lecture Notes in Computer Science #6484*, December 2010.

[XCL10] L. Xia, V. Conitzer, and J. Lang. Aggregating preferences in multi-issue domains by using maximum likelihood estimators. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 399–406. International Foundation for Autonomous Agents and Multiagent Systems, May 2010.

[XCL11] L. Xia, V. Conitzer, and J. Lang. Strategic sequential voting in multi-issue domains and multiple-election paradoxes. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pages 179–188. ACM Press, 2011.

[XLC11] L. Xia, J. Lang, and V. Conitzer. Hypercubewise preference aggregation in multi-issue domains. In *Proceedings of the 22st International Joint Conference on Artificial Intelligence*, pages 158–163. AAAI Press, July 2011.

Edith Hemaspaandra
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623, USA
Email: `eh@cs.rit.edu`

Lane A. Hemaspaandra
Department of Computer Science
University of Rochester
Rochester, NY 14627, USA
Email: `lane@cs.rochester.edu`

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