THE COMPLEXITY OF MANIPULATING k-APPROVAL ELECTIONS*

Andrew Lin
Golisano College of Computing and Information Sciences, Rochester Intitute of Technology, Rochester, NY 14623, U.S.A.

Keywords: Elections manipulation approval complexity.

Abstract: An important problem in computational social choice theory is the complexity of undesirable behavior among agents, such as control, manipulation, and bribery in election systems, which are tempting at the individual level but disastrous for the agents as a whole. Creating election systems where the determination of such strategies is difficult is thus an important goal.

An interesting set of elections is that of scoring protocols. Previous work in this area has demonstrated the complexity of misuse in cases involving a fixed number of candidates, and of specific election systems on unbounded number of candidates such as Borda. In contrast, we take the first step in generalizing the results of computational complexity of election misuse to cases of infinitely many scoring protocols on an unbounded number of candidates.

We demonstrate the worst-case complexity of various problems in this area, by showing they are either polynomial-time computable, NP-hard, or polynomial-time equivalent to another problem of interest. We also demonstrate a surprising connection between manipulation in election systems and some graph theory problems.

1 INTRODUCTION

Election systems are means for aggregating the preferences of individuals to arrive at a decision that attempts to maximize the collective welfare of the individuals. Situations needing such means can arise in political science, such as choosing leaders or writing laws, or also in computation, where a group of agents must agree on an action, such as choose leaders in a parallel algorithm.

An early unfortunate result in computational social choice theory is that all reasonable election systems are subject to misuses by the voters, termed manipulation. Manipulation occurs when one or more agents report insincere preferences for their own benefit. A common manipulation involves burying ones 2^{nd} preference in the preference ordering.

Some election systems also encourage bribery, in which an outside agent can convince, or bribe, some of the agents to change their votes. This is a weakness if the number of affected voters needed to alter the outcome is small.

It is also possible for the administrators of an election to control the outcome of an election by manipulating the set of voters or candidates (Bartholdi et al., 1992). Ways to control the election include encouraging or discouraging potential voters or candidates from participating, or partitioning the voter or candidate set, creating a multiple-round election.

Although it is desirable to design election systems such that the potential for such exploitation is eliminated, several notable results (Gibbard, 1973; Satterthwaite, 1975; Duggan and Schwartz, 2000) have demonstrated that it is impossible to do so, and that every interesting election is subject to manipulation under some conditions.

It is challenged (Bartholdi et al., 1989) that the inevitable manipulation only constitutes a threat when it is computationally easy, i.e. polynomial-time computable, for one to determine for an election system. Much work (Conitzer et al., 2002; Faliszewski et al., 2006) has been done to characterize the worst-case complexities of manipulating different election systems.

We examine an infinite set of election systems, approval-based families of scoring protocols, where each candidate approves of some function $0 \leq f(m) \leq m$ of the m candidates in the election, evaluating the worst-case complexities of various forms of attack. Essentially, we make the first attempt at extending the

*Supported in part by NSF grant IIS-0713061

Lin A..
THE COMPLEXITY OF MANIPULATING Îž-APPROVAL ELECTIONS.
DOI: 10.5220/0003168802120218
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 212-218
ISBN: 978-989-8425-41-6

work of (Hemaspaandra and Hemaspaandra, 2007; Bartholdi et al., 1992) to scoring protocols of an unbounded number of candidates, by looking at an infinite set of election systems. A characterization of an infinite set of election systems is also evaluated in (Faliszewski et al., 2008). A more detailed description of some proofs in this paper can be found in the archived version (Lin, 2010).

We show how some election systems lend themselves to manipulation by simple greedy algorithms, while some are provably hard by reductions from Set Cover or Hitting Set problems. We hope that from our work one can gain a better understanding of what properties of elections encourage or discourage manipulation.

1.1 Our Results

We summarize the most important complexity results in this paper in the tables below, with new results in bold.[1] All results are for constructive misuse.

1.1.1 Unweighted Cases

	1-app	2-app	3-app	k-app, $k \geq 4$
Manip	P	**P**	**P**	**P**
Bribery	P	**P**	**NPC**	**NPC**
CCAV	P	**P**	**P**	**NPC**
CCDV	P	**P**	**NPC**	**NPC**
CCAC	NPC	**NPC**	**NPC**	**NPC**
CCDC	NPC	**NPC**	**NPC**	**NPC**

	1-veto	2-veto	3-veto	k-veto, $k \geq 4$
Manip	P	**P**	**P**	**P**
Bribery	P	**P**	**P**	**NPC**
CCAV	P	**P**	**NPC**	**NPC**
CCDV	P	**P**	**P**	**NPC**
CCAC	NPC	**NPC**	**NPC**	**NPC**
CCDC	NPC	**NPC**	**NPC**	**NPC**

1.1.2 Weighted Voter Cases

	1-app	2-app	3-app	k-app, $k \geq 4$
Manip	P	NPC	NPC	NPC
Bribery	P	NPC	NPC	NPC
CCAV	P	**P**	**sbw**	**NPC**
CCDV	P	**sbw**	**NPC**	**NPC**
CCAC	NPC	**NPC**	**NPC**	**NPC**
CCDC	NPC	**NPC**	**NPC**	**NPC**

[1] In the table, swb is the complexity of Simple Weighted b-Edge Cover of Multigraphs and sbw that of Simple b-Edge Weighted Cover of Multigraphs. CCAV=constructive control by adding voters, CCDV=constructive control by deleting voters, CCAC=constructive control by adding candidates, CCDC=constructive control by deleting candidates

	1-veto	2-veto	3-veto	k-veto, $k \geq 4$
Manip	NPC	NPC	NPC	NPC
Bribery	NPC	NPC	NPC	NPC
CCAV	P	**sbw**	**NPC**	**NPC**
CCDV	P	**P**	**sbw**	**NPC**
CCAC	NPC	**NPC**	**NPC**	**NPC**
CCDC	NPC	**NPC**	**NPC**	**NPC**

1.1.3 Unweighted $Bribery Cases

	1-app	2-app	3-app	k-app, $k \geq 4$
$Bribery	P	**swb**	**NPC**	**NPC**

	1-veto	2-veto	3-veto	k-veto, $k \geq 4$
$Bribery	P	**P**	**swb**	**NPC**

2 PRELIMINARIES

2.1 Definitions and Notations of Election Systems

An election $E = (C,V)$ is defined as a pair, containing m candidates $C = \{c_1,\dots,c_m\}$ and n voters $V = \{v_1,\dots,v_n\}$. Each voter presents a preference profile over the candidates given as a linear order.

In some cases, each voter v may have a predetermined integer weight, $w(v)$. A vote of weight $w(v)$ is counted as $w(v)$ votes in an unweighted election. Additionally, in problems involving bribery, it is also possible for voter v to also have a pre-determined price, $\pi(v)$, which the briber must pay to modify the preferences of the voters, and the goal is to find the cheapest bribery.

An election system, $\mathcal{E}$, specifies how the preferences of the voters over the candidates are aggregated to arrive at a winner or set of winners.

Scoring protocols are a class of election systems, and encompass a large number of common systems such as plurality, veto, Borda count, and some approval-based systems for a fixed number of candidates. We formally define scoring protocols and some examples here.

Definition 1. *Let $E = (C,V)$ be an election of m candidates. A* **scoring protocol** *is an integer vector $(\alpha_1,\dots,\alpha_m)$ with $\alpha_1 \geq \dots \geq \alpha_m \geq 0$ such that each candidate c receives α_i points for each voter that ranks c as their i^{th} favorite alternative.*

A **family of scoring protocols** *is an infinite series $(\alpha^1,\dots,\alpha^m,\dots)$ of scoring protocols where $\alpha^m = (\alpha^m_1,\dots,\alpha^m_m)$ is a scoring protocol of m candidates.*

In **plurality**, each voter gives one point to his or her favorite candidate, whereas in **veto**, each voter vetoes one candidate. In **k-approval**, each voter gives his or her k favorite candidates 1 point. Similarly, in **k-veto**, each voter vetoes his or her least k favorite candidates.

Although not a scoring protocol, another common election system is **approval**. In approval voting, each voter can approve as many or as few voters as he/she chooses. In (Hemaspaandra et al., 2007), it is shown that approval is resistant to many forms of misuse involving the voter set, even with unweighted voters. We will in fact examine the properties of this election system that give rise to the resistance.

A generalization of k-approval and k-veto, **f(m)-approval**, where f is a function of the number of candidates m, is an election where each voter gives 1 point to each of his or her $f(m)$ favorite candidates. In this case, we will assume that $f(m)$ is polynomial-time computable with respect to the quantity m. Absent such assumption, such problems could be difficult simply because we cannot determine the outcome of a manipulation itself.

2.2 Problems of Interest in Computational Social Choice Theory

In this paper, we study the complexity of manipulation, bribery, and control of k-approval and k-veto elections, which have been shown to have some advantages over plurality (Brams and Herschbach, 2001). For each case, there are two problems: constructive, where the goal is to ensure victory of a specific candidate, and destructive, to ensure defeat of such. In manipulation, we attempt to reach this goal by giving preferences to a set of unestablished voters, whereas in bribery we do so by changing the preferences of some established voters. There are two types of control problems: One can alter either the voter set (i.e., by adding or deleting) or the candidate set. In some cases, voters may have weights and costs. We make the formal definitions in this section.

Name. *Constructive $\mathcal{E}$-Control by Adding Voters*

Instance. A set C of candidates, a set V of established voters, and V' of unestablished voters such that $V \cap V' = \emptyset$, distinguished candidate p, and non-negative integer quota q.

Question. Is there a subset $V'' \subseteq V'$ with $||V''|| \leq q$ such that p is a winner of the $\mathcal{E}$ election with candidate set C and voter set $V \cup V''$?

Another problem of interest is Constructive $\mathcal{E}$-Control by Deleting Voters, in which we ensure the victory of p by deleting at most q voters.

Name. *Constructive $\mathcal{E}$-Control by Adding Candidates*

Instance. A set C of candidates, a set V of established voters, and C' of unestablished candidates such that $C \cap C' = \emptyset$, distinguished candidate p, and non-negative integer quota q. In this case, V provides a linear preference ordering over $C \cup C'$.

Question. Is there a subset $C'' \subseteq C'$ with $||C''|| \leq q$ such that p is a winner of the $\mathcal{E}$ election with candidate set $C \cup C''$ and voter set V?

$\mathcal{E}$-Control by Deleting Candidates is defined similarly.

In manipulation, a subset of voters working together seek to determine the outcome of an election.

Name. *Constructive $\mathcal{E}$-Manipulation*

Instance. A set C of candidates, a set V of established voters, and V' of unestablished voters such that $V \cap V' = \emptyset$, and distinguished candidate p.

Question. Is there an assignment of preference profiles for V' such that p is a winner of the $\mathcal{E}$ election with candidate set C and voter set $V \cup V'$?

In all of these problems, another problem of interest is the cases where voters have weights. In such case, we denote the problem by $\mathcal{E}$-weighted-constructive-manipulation and $\mathcal{E}$-weighted-destructive-manipulation.

Name. *Constructive $\mathcal{E}$-Bribery*

Instance. A set C of candidates, a set V of voters, distinguished candidate p, and non-negative integer quota q.

Question. Is it possible to make p a winner of the $\mathcal{E}$ election by changing the preference profiles of at most q voters in V.

As in the cases of control and manipulation, bribery is also defined for cases of weighted voters. We denote this problem $\mathcal{E}$-Weighted-Bribery. In addition, each voter can be assigned a price tag of a non-negative integer. In this case, q is our budget, and we want to achieve our goal (constructive or destructive bribery) by spending at most total cost q in our bribery. We denote this problem by $\mathcal{E}$-\$Bribery and $\mathcal{E}$-Weighted-\$Bribery if voters have both weights and prices.

An election system $\mathcal{E}$ is **resistant** to misuse if determining if such a misuse exists for a given election is NP-hard. It is **vulnerable** if such a decision is polynomial-time computable.

2.3 Some Important NP-Complete Problems

Common NP-complete problems of choice for showing NP-hardness of election systems include versions of Set Cover, Knapsack, and Hitting Set.

In Exact 3-Set Cover (X3C) (Karp, 1972) (See also (Garey and Johnson, 1979, problem SP2)), a set $S = \{s_1, \ldots, s_{3m}\}$ and subsets $T_1, \ldots, T_n \subseteq S$ such that $||T_i|| = 3$ are given. We wish to find m such subsets that exactly cover S. Reductions from similar problems were used in (Brelsford et al., 2008) to show the resistance of the Borda count election to bribery.

In Hitting Set, a problem used to show hardness of some manipulations in the candidate set, we are given a set $S = \{s_1, \ldots, s_m\}$, n subsets of S, $T_1, \ldots, T_n$, and positive integer $1 \leq q \leq m$. We wish to determine if there is a subset of q elements of S that hit each of the n given subsets. This problem is of interest in the issue of control by adding and deleting candidates (Bartholdi et al., 1989).

2.4 Edge Cover, b-Edge Cover and their Relationship to Approval-based Elections

Among other results, we demonstrate the relationship between some approval-based election systems and problems of edge coverings in graphs. In general, this occurs in systems that distinguish two candidates from the remaining as edges connect pairs of vertices. The well known problem of Edge Cover, which is polynomial-time computable, is defined below.

Definition 2. *An **edge cover** of a graph is a set of edges such that every vertex of the graph is incident to at least one edge of the set. In the decision problem **Edge Cover**, we are given a graph $G = (V,E)$ and positive integer q, and wish to determine if there exists an edge cover $C \subseteq E$ for G of at most q edges.*

In the variation b-Edge Cover, each vertex v is to be covered by a minimum of some number, $b(v)$, of edges. There are several interesting versions: Each edge can be chosen only once (Simple b-Edge Cover), an arbitrary number of times (b-Edge Cover), or have a capacity and be chosen up to that many times (Capacitated b-Edge Cover) (see (Schrijver, 2003) 34.1, 34.7, 34.8), all of which are polynomial-time computable (Pulleyblank, 1973; Cunningham and III, 1978; Gabow, 1983; Anstee, 1987).

Theorem 1. *Simple b-Edge Cover for Multigraphs, in which we are given a multigraph G with b-values $b(v)$ defined for each vertex $v \in V(G)$ and integer q, and wish to determine if G can be simple b-edge covered by at most q edges, is polynomial-time computable.*

This follows from Capacitated b-Edge Cover (Schrijver, 2003), with edge capacity corresponding to the number of edges between each two vertices.

We introduce two additional versions of b-Edge Cover of Multigraphs, involving weights. These problems are of interest in some weighted election manipulations.

In **Simple Weighted b-Edge Cover of Multigraphs**, a weight is assigned to each edge in the multigraph. We are interested in whether a cover exists of at most weight q.

In **Simple b-Edge Weighted Cover of Multigraphs**, a weight is assigned to each edge. We are interested in finding a cover $C \subseteq E$ of at most q vertices such that each vertex $v \in V$ is incident to edges in C of total weight at least $b(v)$.

We introduce these two problems to convert the otherwise obscure problems of election manipulation to that of more natural graph theory-related problems. It is hoped that algorithms and heuristics available for similar problems can be applied toward these two versions, and thus to some problems in manipulation. The complexity of either of these variations is left as an open problem.

3 MISUSES OF APPROVAL BASED ELECTIONS IN UNWEIGHTED CASES

We examine the problems of manipulation, bribery, and control, for $f(m)$-approval elections, and also for the special cases of k-approval and k-veto, in unweighted elections. This extends the known results of 1-approval and 1-veto (Bartholdi et al., 1989; Faliszewski et al., 2006). We wish to examine how the difficulty of these problems increase as we generalize the scoring protocol.

3.1 Destructive Misuses

In several previous results (Hemaspaandra and Hemaspaandra, 2007; Russell, 2007), it has been shown that destructive misuse is vastly easier than constructive misuse for many election systems. The principle is as follows: To keep p from winning the election, it suffices to ensure some candidate $p' \neq p$ beats p. However, this can occur iff bribing voters giving p the greatest lead against p' suffices.

We conclude that all unweighted $f(m)$-approval elections are vulnerable to destructive bribery, manipulation, and control by adding or deleting voters.

3.2 Manipulation of Approval-Based Scoring Protocols

We show that manipulation of all families of scoring protocols of the form $f(m)$-approval is easy. This observation was made independently by Procaccia (Procaccia, 2009).

Theorem 2. *Unweighted $f(m)$-approval elections are vulnerable to constructive manipulation.*

Proof. Consider an $f(m)$-approval election $E = (V, C)$, distinguished candidate $p \in C$, and set of non-established voters V'. A simple greedy algorithm that finds a constructive manipulation is to iteratively approve p and the $f(m) - 1$ non-distinguished candidates currently with the lowest scores for each voter in V'.

Correctness of this algorithm can be shown by induction on $||V'||$, by proving that a manipulation exists iff there exists one such that at least one voter approves of p and the $f(m) - 1$ non-distinguished candidates currently with the lowest scores. □

3.3 Bribery in Approval-based Scoring Protocols

The goal of a briber is to determine whether it is possible to ensure victory of a desired candidate by altering voter preferences. Unweighted 1-approval and 1 veto elections are known to be vulnerable to unpriced constructive bribery, by simple greedy algorithms (Faliszewski et al., 2006).

Theorem 3. *Unweighted* 2*-veto elections are vulnerable to unpriced constructive bribery.*

Proof. Consider a 2-veto elections. We iteratively bribe a voter v vetoing $\{p, c\}$ such that no other voter v' vetoes $\{p, c'\}$ such that c currently has fewer vetoes than c', and give the vetoes to the candidates with currently the fewest vetoes. Correctness is shown by induction on the bribery size. □

Our next result shows the connection between Set Cover and approval-based elections.

Theorem 4. *Unweighted k-approval elections for $k \geq 3$ and k-veto for $k \geq 4$ are resistant to unpriced constructive bribery.*

Proof. Consider the following reduction from X3C to bribery in 3-approval elections.

Let $S = \{s_1, \ldots, s_{3m}\}$ and $T_1, \ldots, T_n$ be 3-subsets of S. Consider the following 3-approval election of $C = \{p, p', p''\} \cup \{s_1, \ldots, s_{3m}\} \cup \{b_1, \ldots, b_{6(nm+m-n)}\}$.

For each set $T_i = \{t_{i,1}, t_{i,2}, t_{i,3}\}$, voter T_i approves $t_{i,1}, t_{i,2}, t_{i,3}$, and for $1 \leq i \leq 3(nm + m - n)$ voter S_i approves b_{2i-1}, b_{2i}, and one of $s_1, \ldots, s_{3m}$, such that each s_j receives exactly $n+1$ approvals. Finally, $n - m$ voters approve p, p', p''. We set our bribery quota q to m.

A successful bribery must remove one point from each candidate of S, corresponding to a solution to X3C. Buffer candidates can be added for cases of k-approval for $k \geq 4$. For 4-veto elections we veto p and the three candidates corresponding to each subset. □

Theorem 5. *Unweighted* 2*-approval and* 3*-veto elections are vulnerable to unpriced constructive bribery.*

Proof. Consider bribery in 2-approval elections. Clearly, we only bribe voters not approving p to approve p.

For $c \in C$, let $\text{Approvals}(c)$ be the number of voters currently approving c. Following bribery, p will receive $\text{Approvals}(p) + q$ approvals. Also define $\text{Deficit}(c) = \text{Approvals}(p) + q - \text{Approvals}(c)$, as the number of excess approvals p ends with relative to c.

Consider the following instance of Simple b-Edge Cover of Multigraphs. Let $X = \sum_{c \neq p} \max(0, -\text{Deficit}(c))$ be the number of "excess" approvals that must be removed. If $X > 2q$, then bribery is not possible. Similarly, let $D = \sum_{c \neq p} \max(0, \text{Deficit}(c))$ be the number of approvals that we can give to the non-distinguished candidates. We can bribe at most $\min(D, q)$ voters.

Construct G as follows. Let $V(G) = C \backslash \{p\}$. For every voter approving u and v such that $p \notin \{u, v\}$, we add edge (u, v). We set the b-values to $b(v) = \max(0, -\text{Deficit}(c))$, and our covering quota to $q' = \min(D, q)$. We can show that G has a simple b-edge covering of $q' = \min(D, q)$ edges iff there exists a bribery of q voters making p a winner.

In a similar construction for 3-veto elections, votes vetoing candidates p, c_1, and c_2 correspond to an edge between c_1 and c_2 in G. □

3.4 Controlling an Election via Voters

Algorithms and reductions similar to cases of bribery can be applied to control by adding or deleting voters.

Theorem 6. *Unweighted* 1 *and* 2*-approval and* 1 *and* 2*-veto elections are vulnerable to control by adding or deleting voters.* 3*-approval is also vulnerable to control by adding voters, and* 3*-veto to deleting voters.*

Proof. Unweighted 3-approval and 2-veto elections are vulnerable to constructive control by adding voters, and 2-approval and 3-veto elections to deleting voters using Simple b-Edge Cover of Multigraphs. The others are due to greedy algorithms. □

Theorem 7. *Unweighted k-approval elections for $k \geq 4$ and k-veto for $k \geq 3$ are resistant to constructive control by adding voters, and k-approval for $k \geq 3$ and k-veto for $k \geq 4$ by deleting voters, due to reduction from X3C.*

3.5 Controlling an Election via Candidates

Reductions from Hitting Set (Bartholdi et al., 1992; Hemaspaandra et al., 2007) have shown that plurality and veto are resistant to both constructive and destructive control by adding and deleting candidates. The reductions are from Hitting Set, and buffer candidates can be added for the cases of *k*-approval and *k*-veto. These systems are thus resistant to control by adding or deleting candidates.

4 ON WEIGHTED AND PRICED CASES OF ELECTION MISUSE

From the results in (Hemaspaandra and Hemaspaandra, 2007), we conclude that 1-approval is the only approval-based family of scoring protocol that is vulnerable to weighted manipulation and weighted bribery.

Theorem 8. *Unweighted* 1*-approval,* 1*-veto, and* 2*-veto elections are vulnerable to constructive $Bribery (i.e., bribery with priced voters).*

Proof. The case of $Bribery for 1-approval is shown in (Faliszewski et al., 2006). In 1-veto, we iteratively bribe the cheapest voters vetoing p, giving the veto to the candidate currently having the fewest voters.

In the case of 2-veto, we will bribe only voters vetoing p, and for each candidate $c \neq p$, we will bribe the cheapest voters vetoing $\{p, c\}$. For each $c \in C$, let $\text{Vetoes}(c)$ be the total number of vetoes given to c initially.

Suppose that a total of s voters are bribed, so that p ends with $\text{Vetoes}(p) - s$ vetoes. Let $C_1 = \{c \in C \backslash \{p\} | \text{Vetoes}(c) < \text{Vetoes}(p) - s\}$ and $C_2 = \{c \in C \backslash \{p\} | \text{Vetoes}(c) \geq \text{Vetoes}(p) - s\}$. We need to make sure that each candidate $c \neq p$ ends up with at least $Vetoes(p) - s$ vetoes.

Let $s = s_1 + s_2 + s_3$ such that s_1 of the bribed voters veto p and a candidate in C_1 and $s_2 + s_3$ voters veto p and a candidate in C_2, such that at least s_2 voters veto $\{p, c\}$ where no more than $\text{Vetoes}(c) - (\text{Vetoes}(p) - s)$ voters veto $\{p, c\}$.

For each case, we choose the cheapest s_1 voters vetoing p and a candidate in C_1. We then iteratively choose the cheapest s_2 voters vetoing p and a candidate in C_2, such that for each $c \in C_2$, no more than $\text{Vetoes}(c) - (\text{Vetoes}(p) - s)$ voters vetoing $\{p, c\}$ are chosen. We finally choose the cheapest s_3 remaining voters vetoing p and a candidate in C_2 to bribe. We try this for each $s_1 + s_2 + s_3 \leq ||V||$. □

Theorem 9. *Weighted* 1*-approval and* 1*-veto elections are vulnerable to constructive control by adding and deleting voters.* 2*-approval is vulnerable to adding voters, and* 2*-veto to deleting voters.*

Proof. The cases of 1-approval and 1-veto follow from greedy algorithms, for example, iteratively adding the heaviest voters approving p.

Consider an addition of voters in a weighted 2-approval election that makes p a winner. Clearly we only add voters approving p. Let v be an unestablished voter that approves of $\{p, c\}$ such that no other unestablished voter v' approves $\{p, c'\}$ with c' having more approvals initially than c. We may assume that at least one voter approving of $\{p, c\}$ is added. It also suffices to add the heaviest voter approving $\{p, c\}$. A correct algorithm is to add such voters iteratively. □

The cases of constructive control by adding (deleting) voters in 3-approval and 2-veto (2-approval and 3-veto), as well as priced bribery in 2-approval and 3-veto, are polynomial-time equivalent to some weighted variations of b-Edge Cover as follows: The weights of voters correspond to the weighted coverings, while the prices in bribery problems correspond to weighted edges. The complexity of these two problems is left as an open problem.

5 RESULTS AND DISCUSSION

These results show the variance of complexity in misuse among different problems in election systems in elections of the form *k*-approval, *k*-veto, and $f(m)$-approval, and give the first results of complexity for infinitely many scoring protocols of an unbounded number of candidates. There are a few interesting cases: These manipulations can either be easy by a simple greedy algorithm, equivalent to a corresponding variation of b-Edge Cover, which is easy for the unweighted and unpriced cases but unknown for the weighted or priced variations, or hard by reduction from Set Cover. Manipulations involving the candidate sets are always difficult as a result of Hitting Set.

These results demonstrate the strengths and weaknesses of approval-based election systems, and we hope these results can lead to further generalizations and possible developments of systems that better resist such attacks.

It is important to realize that NP-completeness only addresses the worst-case complexity of a given problem, and does not take into consideration the distribution of problems that might be given. Some simple distributions were considered in (Walsh, 2009; Friedgut et al., 2008), and it may be of interest to characterize the complexity of more interesting and realistic distributions, depending upon the application..

This model also makes the assumption that in a k-approval election, each voter may vote for any combination of the k candidates independently. We know that in practice, most elections do not follow this principle. It may thus be of interest to characterize these properties in a more realistic distribution of voter preferences.

ACKNOWLEDGEMENTS

We wish to offer our special thanks to Dr. E. Hemaspaandra for pointing out the connection between b-Edge Cover and elections as well as proofreading.

REFERENCES

Anstee, R. (1987). A polynomial algorithm for b-matchings: an alternative approach. *Information Processing Letters*, pages 554–559.

Bartholdi, J., Tovey, C., and Trick, M. (1989). The computational difficulty of manipulating an election. *Social Choice and Welfare*, pages 227–241.

Bartholdi, J., Tovey, C., and Trick, M. (1992). How hard is it to control an election? *Mathematical and Computer Modelling*, pages 27–40.

Brams, S. and Herschbach, D. (2001). The science of elections. In *Science*, page 1449.

Brelsford, E., Faliszewski, P., Hemaspaandra, E., Schnoor, H., and Schnoor, I. (2008). Approximability of manipulating elections. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 44–49.

Conitzer, V., Lang, J., and Sandholm, T. (2002). When are elections with few candidates hard to manipulate? *Journal of the ACM, Volume 54, Issue 3, Article 14*, pages 1–33.

Cunningham, W. and III, A. M. (1978). A primal algorithm for optimum matching. *Polyhedral Combinatorics, Mathematical Programming Study 8*, pages 50–72.

Duggan, J. and Schwartz, T. (2000). Strategic manipulability without resoluteness or shared beliefs: Gibbard-satterthwaite generalized. *Social Choice and Welfare*, pages 85–93.

Faliszewski, P., Hemaspaandra, E., and Hemaspaandra, L. (2006). How hard is bribery in elections. *Journal of AI Research, Volume 35*, pages 485–532.

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

Friedgut, E., Kalai, G., and Nisan, N. (2008). Elections can be manipulated often. In *Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 243–249.

Gabow, H. (1983). An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computation*, pages 448–456.

Garey, M. and Johnson, D. (1979). Computers and intractability: A guide to the theory of NP-completeness. *W.H. Freeman and Company*.

Gibbard, A. (1973). Manipulation of voting schemes: a general result. *Econometrica*, pages 587–601.

Hemaspaandra, E. and Hemaspaandra, L. (2007). Dichotomy for voting systems. *Journal of Computer and System Sciences*, pages 73–83.

Hemaspaandra, E., Hemaspaandra, L., and Rothe, J. (2007). Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, pages 255–285.

Karp, R. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103.

Lin, A. (2010). The complexity of manipulating k-approval elections,arxiv:1005.4159.

Procaccia, A. (2009). Personal communication.

Pulleyblank, W. (1973). Faces of matching polyhedra. *Ph.D. Thesis, Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario*.

Russell, N. (2007). Complexity of control of Borda count elections. *Rochester Institute of Technology*.

Satterthwaite, M. (1975). Vote elicitation: Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory 10 (April 1975)*, pages 187–217.

Schrijver, A. (2003). Combinatorial optimization. *Springer*.

Walsh, T. (2009). Where are the really hard manipulation problems? the phase transition in manipulating the veto rule. In *International Joint Conference on Artificial intelligence*, pages 324–329.