Cloning in Elections[1]

Edith Elkind, Piotr Faliszewski, and Arkadii Slinko

Abstract

We consider the problem of manipulating elections via cloning candidates. In our model, a manipulator can replace each candidate c by one or more *clones*, i.e., new candidates that are so similar to c that each voter simply replaces c in his vote with the block of c's clones. The outcome of the resulting election may then depend on how each voter orders the clones within the block. We formalize what it means for a cloning manipulation to be successful (which turns out to be a surprisingly delicate issue), and, for a number of prominent voting rules, characterize the preference profiles for which a successful cloning manipulation exists. We also consider the model where there is a cost associated with producing each clone, and study the complexity of finding a minimum-cost cloning manipulation. Finally, we compare cloning with the related problem of control via adding candidates.

1 Introduction

In real-life elections with more than two candidates, the winner does not always have broad political support. This is possible, for example, when the opposing views are represented by several relatively similar candidates, and therefore the vote in favor of the opposition gets "split". For example, it is widely believed that in the 2000 U.S. Presidential election spoiler candidate Ralph Nader have split votes away from Democratic candidate Al Gore allowing Republican candidate George W. Bush to win.

One can also imagine scenarios where having several similar candidates may bias the outcome in their favor. For example, suppose that an electronics website runs a competition for the best digital camera by asking consumers to vote for their two favorite models from a given list. If the list contains one model of each brand, and half of the consumers prefer Sony to Nikon to Kodak, while the remaining consumers prefer Kodak to Nikon to Sony, then Nikon will win the competition. On the other hand, if each brand is represented by several similar models, then the "Sony" customers are likely to vote for two models of Sony, the "Kodak" customers are likely to vote for two models of Kodak, and Nikon receives no votes.

The above-described scenarios present an opportunity for a party that is interested in manipulating the outcome of an election. Such a party—most likely, a campaign manager for one of the candidates—may invest in creating "clones" of one or more candidates in order to make its most preferred candidate (or one of its "clones") win the election. A natural question, then, is which voting rules are resistant to such manipulation, and whether the manipulator can compute the optimal cloning strategy for a given election.

The first study of cloning was undertaken by Tideman [18], who introduced the concept of "independence of clones" as a criterion for voting rules. He considered a number of well-known voting rules, and discovered that among these rules, STV was the only one that satisfied this criterion. However, STV does not satisfy many other important criteria for voting rules, e.g., Condorcet consistency. Thus, Tideman [18] proposed a voting rule, the "ranked pairs rule," that was both Condorcet-consistent and independent of clones in all but a small fraction of settings. Subsequently, Zavist and Tideman [19] proposed a modification of this rule that is completely independent of clones. Later it was shown that some other voting rules, such as Schulze's rule [17], are also resistant to cloning.

[1] This paper in its preliminary form will be presented at AAAI-2010.

A related concept of composition consistency as well as its weaker version, cloning consistency, was considered by Laffond at al. [11] and by Laslier [12]. They proved that a number of tournament solutions such as the Banks Set, the Uncovered Set, the Tournament Equilibrium Set (TEQ), and the Minimal Covering Set are composition-consistent. They also demonstrated that various tournament solution concepts and voting rules such as the Top Cycle, the Slater rule, the Copeland rule, and all scoring rules are not composition-consistent.

In this paper we take a rather different perspective on cloning: Instead of looking at cloning as a manipulative action that should be prevented, we view cloning as a campaign management tool. This point of view raises a number of questions that have not been considered before(or, have not been considered from this perspective):

What does it mean for cloning to be successful? The campaign manager can produce clones of existing candidates, but the voters rank them in response. We assume that clones are similar enough to be ranked as a group by each voter; however, the order of clones in such groups is specific to a particular voter. Since the campaign manager cannot control or predict the order of clones in each voter's ranking, we assume that this order is random (that is, each voter assigns equal probability to each possible order of the cloned candidates). Thus, the success of a cloning manipulation is a random event, and we can measure it probability. Let q be some real number between 0 and 1. We say that manipulation by cloning is *q-successful* if the probability of electing the desired candidate is at least q. We focus on two extreme cases: one where no matter what the voters do, the campaign manager's preferred candidate p wins (cloning is 1-successful), and one where there is a non-zero chance that p wins (by a slight abuse of notation, we will call such cloning is 0-successful).

In which instances of elections can cloning be successful? While previous work demonstrates that many well-known voting rules are susceptible to cloning, no attempt has been made to characterize the elections in which a specific candidate can be made a winner with respect to a given voting rule by means of cloning. However, from the point of view of a campaign manager who considers cloning as one of the ways to run the campaign, such characterizations are crucial. Thus, in this paper we characterize cloning-manipulable elections for several prominent voting rules. Often, manipulable elections can be characterized in terms of well-known notions of social choice such as Pareto optimality, Condorcet loser, or Uncovered Set.

Which candidates can be cloned and to what extent? The existing work on cloning does not place any restrictions on the number of clones that can be introduced, or on which candidates can be cloned at all. On the other hand, it is clear that in practical campaign management scenarios these issues cannot be ignored: not all candidates can be cloned, and creating a clone of a given candidate may be costly. Thus, we consider settings in which each clone of each candidate comes at some cost, and we seek a least expensive successful cloning strategy. However, mostly we focus on the standard model where clones come at zero cost, and on the *unit cost* model, where all clones have the same cost.

What is the computational complexity of finding cloning strategies? Finally, we consider the computational complexity of finding successful cloning strategies. In practice, it is not sufficient to know that cloning *might* work: We need to know exactly which strategy to use. We believe that our paper is the first to consider the computational aspect of cloning. Following the line of work initiated by the seminal papers of Bartholdi, Tovey, and Trick [1, 2], we seek to establish which cloning problems are NP-hard for a given voting rule, and which are solvable in polynomial time.

One might argue that in real-life elections cloning isn't really a practical campaign management tool. After all, creating even a single clone may well be too difficult or too costly. Nonetheless, below we provide two natural examples where our model of cloning is practical and well-motivated.

First, let us consider an election in which parties nominate candidates for some position, and each party can nominate several candidates. From the point of view of the voters, especially those not following the political scene closely, candidates from the same party are perceived as clones. A party's campaign manager might attempt to strategically choose the number of candidates her party should nominate, and, in fact, she might even be able to affect the number of candidates nominated by other parties (e.g., by accusing them of not giving the voters enough choice).

Second, let us consider an environment where, as suggested by Ephrati and Rosenschein in their classic paper [6], software agents vote to choose a joint plan (that is, the candidates are possible joint plans or steps of possible joint plans). In such a system, the agents can easily come up with minor variations of the (steps of the) plan, effectively creating clones of the candidates. (Laslier [12] has given a very similar example regarding a society of agents choosing a project to implement.) In both cases, the assumption that all clones are ranked contiguously and the requirement that finding a successful cloning strategy should be computationally easy are particularly relevant and realistic.

2 Preliminaries

Given a set A of *alternatives* (also called *candidates*), a voter's *preference* R is a *linear order* over A, i.e., a total transitive antisymmetric binary relation over A. An *election* E with n voters is given by its set of alternatives A and a *preference profile* $\mathcal{R} = (R_1, \dots, R_n)$, where R_i is the preference of voter i; we write $E = (A, \mathcal{R})$. For readability, we sometimes write $\succ_i$ in place of R_i. Also, we denote by $|\mathcal{R}|$ the number of voters in the election.

A voting rule $\mathcal{F}$ is often defined as a mapping from elections with a fixed set of alternatives A to the set 2^A of all subsets of A. However, in this work, we are interested in situations where the number of alternatives may change. Thus, we require voting rules to be defined for arbitrary finite sets of alternatives and preference profiles over those alternatives. Most well-known voting rules (see below) fit this more demanding definition; for ones that do not (e.g., scoring rules), we explain how to adapt their standard definition to our setting. Thus, we say that a *voting rule* $\mathcal{F}$ is a mapping from pairs of the form $E = (A, \mathcal{R})$, where A is some finite set and $\mathcal{R}$ is a preference profile over A, to subsets of A. The elements of $\mathcal{F}(E)$ are called the *winners* of the election E. Thus, we allow an election to have more than one winner, i.e., we work with social choice correspondences (also called *non-unique winner* model.)

In this paper we consider the following voting rules (for all rules described in terms of scores the winners are the alternatives with the maximum score):

Plurality. The *Plurality score* $Sc_P(c)$ of a candidate $c \in A$ is the number of voters that rank c first.

Veto. The *Veto score* $Sc_V(c)$ of a candidate $c \in A$ is the number of voters that do not rank c last.

Borda. Given an election $(A, \mathcal{R})$ with $|\mathcal{R}| = n$, the *Borda score* $Sc_B(c)$ of a candidate $c \in A$ is given by $Sc_B(c) = \sum_{i=1}^{n} |\{a \in A \mid c \succ_i a\}|$.

k-Approval. For any $k \geq 1$, the *k-Approval score* $Sc_k(c)$ of a candidate $c \in A$ is the number of voters that rank c in the top k positions. Plurality is simply 1-Approval.

Plurality with Runoff. In the first stage, all but two candidates with the top two Plurality scores are eliminated. Then the winner is the one of the survivors that is preferred to the other one by at least half of the voters. We may need to break a tie after the first round, if more than one candidate has the best or the second best score; to this end we use the *parallel universes* tie-breaking rule [4].

Maximin. Given an election $(A, \mathcal{R})$ with $|\mathcal{R}| = n$, for any $a, c \in A$, let $W(c, a) = |\{i \mid c \succ_i a\}|$. The *Maximin score* $Sc_M(c)$ of a candidate $c \in A$ is given by $Sc_M(c) = \min_{a \in A} W(c, a)$, i.e., it is the number of votes c gets in his worst pairwise contest.

Copeland. The *Copeland score* $Sc_C(c)$ of a candidate $c \in A$ is $|\{a \mid W(c, a) > W(a, c)\}| - |\{a \mid W(a, c) > W(c, a)\}|$. This is equivalent to saying that for each candidate a, c gets 1 point if she wins the pairwise contest against a, 0.5 point if there is a tie, and 0 if she loses the contest.[2]

Many results of this paper are computational and thus we assume the reader is somewhat familiar with standard notions of computational complexity such as classes P and NP, many-one reductions, NP-hardness and NP-completeness. Our NP-hardness results typically follow by reductions from Exact Cover by 3-Sets problem, defined below.

Definition 2.1 ([9])**.** *An instance $(G, \mathcal{S})$ of* Exact Cover by 3-Sets (X3C) *is given by a ground set $G = \{g_1, \ldots, g_{3K}\}$, and a family $\mathcal{S} = \{S_1, \ldots, S_M\}$ of subsets of G, where $|S_i| = 3$ for each $i = 1, \ldots, M$. It is a "yes"-instance if there is a subfamily $\mathcal{S}' \subseteq \mathcal{S}$, $|\mathcal{S}'| = K$, such that for each $g_i \in G$ there is an $S_j \in \mathcal{S}'$ such that $g_i \in S_j$, and a "no"-instance otherwise.*

3 Our Framework

Cloning and independence of clones were previously defined in [14, 18, 19]. However, we need to modify the definition given in these papers in order to model the manipulator's intentions and the budget constraints. We will now describe our model formally.

Definition 3.1. *Let $E = (A, (R_1, \ldots, R_n))$ be an election with a set of candidates $A = \{c_1, \ldots, c_m\}$. We say that an election $E' = (A', (R'_1, \ldots, R'_n))$ is obtained from E by replacing a candidate $c_j \in A$ with k* clones *for some $k > 0$ if $A' = A \setminus \{c_j\} \cup \{c_j^{(1)}, \ldots, c_j^{(k)}\}$ and for each $i \in [n]$, R'_i is a total order over A' such that:*

(i) for any $a \in A \setminus \{c_j\}$ and any $s \in [k]$ it holds that $c_j^{(s)} \succ'_i a$ if and only if $c_j \succ_i a$;
(ii) for any $a, b \in A \setminus \{c_j\}$ it holds that $a \succ'_i b$ if and only if $a \succ_i b$.

We say that an election $E^* = (A^*, \mathcal{R}^*)$ is cloned *from an election $E = (A, \mathcal{R})$ if there is a vector of non-negative integers $(k_1, \ldots, k_m)$ such that E^* is derived from E by replacing each c_j, $j = 1, \ldots, m$, with k_j clones.*

Thus, when we clone a candidate c, we replace her with a group of new candidates that are ranked together in all voters' preferences. Observe that according to the definition above, cloning a candidate c_j once means simply changing his name to $c_j^{(1)}$ rather than producing an additional copy of c_j. While not completely intuitive, this choice of terminology simplifies some of the arguments in the rest of the paper.

The definition above is essentially equivalent to the one given in [19]; the main difference is that we explicitly model cloning of more than one candidate. However, we still need to

[2]The original Copeland rule [5] was applied to tournaments and the score was the number of wins.

introduce the two other components of our model: a definition of what it means for a cloning to be successful, and the budget.

We start with the former assuming throughout this discussion that the voting rule is fixed. We observe that the final outcome of cloning depends on the relative ranking of the clones chosen by each voter, which is not under the manipulator's control. Thus, a cloning may succeed for some orderings of the clones, but not for others. The election authorities may approach this issue from the worst-case perspective, and consider it unacceptable when a given cloning succeeds for *at least one* ordering of clones by voters. Alternatively, they can take an average-case perspective, i.e.., assume that the voters rank the clones randomly and independently, with each ordering of the clones being equally likely (due to the similarities among the clones), and consider it acceptable for a cloning manipulation to succeed with probability that does not exceed a certain threshold. On the other hand, a (cautious) manipulator would view cloning as successful only if it succeeds for *all* orderings.

Definition 3.2. *Given a positive real* $0 < q \leq 1$*, we say that a manipulation by cloning (or simply cloning) is* q-successful *if (a) the manipulator's preferred candidate is not a winner of the original election, and (b) a clone of the manipulator's preferred candidate is a winner of the cloned election with probability at least* q*.*

The two approaches discussed above are special cases of this framework. Indeed, a cloning succeeds for all orderings if and only if it is 1-successful, and it succeeds for some ordering if and only if it is q-successful for some $q > 0$ no matter how small it is; we abuse notation by referring to such cloning as 0-successful. Saying that cloning is 0-successful is equivalent to saying that the cloning would be successful if the manipulator could dictate each voter how to order the clones. We will use this observation very often as it simplifies proofs.

Observe that, according to our definition, the manipulator succeeds as long as *any one* of the clones of the preferred candidate wins. This assumption is natural if the clones represent the same company (e.g., Coke Light and Coke Zero) or political party. However, if a campaign manager has created a clone of his candidate simply by recruiting an independent candidate to run on a similar platform, he may find the outcome in which this new candidate wins less than optimal. We could instead define success as a victory by the original candidate (i.e., the clone $c^{(1)}$), but, at least for neutral voting rules, this is essentially equivalent to the previous definition. Indeed, any preference profile in which the original candidate wins can be transformed into one in which some clone wins, by switching their order in each voter's preferences so $c^{(1)}$ wins with the same probability as any other clone.

Note that our definition of q-successful cloning is similar in spirit to that of [7], where voters are bribed to increase their probabilities of voting as the briber wants.

Another issue that we need to address is that of the costs associated with cloning. Indeed, the costs are an important aspect of realistic campaign management, as the manager is always restricted by the budget of the campaign. The most general way to model the cloning costs for an election with the initial set of candidates $A = \{c_1, \dots, c_m\}$ is via a *price function* $p\colon [m] \times \mathbb{Z}^+ \to \mathbb{Z}^+ \cup \{0\} \cup \{\infty\}$, where $p(i, j)$ denotes the cost of producing the j-th copy of candidate c_i. Note that $p(i, 1)$ corresponds to not producing additional copies of i, so we require $p(i, 1) = 0$ for all $i \in [m]$. We remark that it is natural to assume that all costs are non-negative (though some of them may equal zero); the assumption that all costs are integer-valued is made for computational reasons. This is not a real restriction as monetary values are discrete.

We assume that for some positive integer t the marginal cost of introducing an additional cloned candidate becomes constant, that is, $p(i, j) = p(i, t)$ for $j > t$. This ensures that the price function is succinctly representable. Thus our cost function is in fact a mapping

$p\colon [m] \times [t] \to \mathbb{Z}^+ \cup \{0\} \cup \{\infty\}$. The two natural special cases of our model defined below—Zero Cost and Unit Cost models—satisfy this condition.

Definition 3.3. *An instance of the q-Cloning problem for $q \in [0,1]$ is given by the initial set of candidates $A = \{c_1, \dots, c_m\}$, a preference profile $\mathcal{R}$ over A, a manipulator's preferred candidate $c \in A$, a parameter $t > 1$, a price function $p\colon [m] \times [t] \to \mathbb{Z}^+ \cup \{0\} \cup \{\infty\}$, a budget B, and a voting rule $\mathcal{F}$. We ask if there exists a q-successful cloning with respect to $\mathcal{F}$ that costs at most B.*

For most voting rules that we consider, it is easy to bound the number of clones needed for 0-successful or 1-successful cloning (if one exists); moreover, this bound is usually polynomial in n and m. We focus on two natural special cases of q-Cloning:

1. Zero Cost (ZC): $p(i,j) = 0$ for all $i \in [m]$, $j \in \mathbb{Z}^+$. In this case we would like to decide whether an election is manipulable at all.

2. Unit Cost (UC): $p(i,j) = 1$ for all $i \in [m]$, $j \geq 2$. This model assumes that creating each new clone has a fixed cost equal for all candidates.

We say that an election E is *q-manipulable by cloning with respect to a voting rule $\mathcal{F}$* if there is a q-successful manipulation by cloning with respect to $\mathcal{F}$ in the ZC model. Further, we say that E is *manipulable by cloning with respect to $\mathcal{F}$* if it is 0-manipulable with respect to $\mathcal{F}$, and *strongly manipulable by cloning with respect to $\mathcal{F}$* if it is 1-manipulable with respect to $\mathcal{F}$.

In the rest of the paper, we discuss the complexity of the q-Cloning problem for a number of well-known voting rules, focusing on the ZC and UC models. Clearly, hardness results for these special cases also imply hardness results for the general model. Somewhat less obviously, hardness results for the ZC q-Cloning imply hardness results for UC q-Cloning: it suffices to set $B = \infty$.

Note that for polynomial-time computable voting rules 0-Cloning is clearly in NP. After a moment's thought, we can also see that q-Cloning for such rules is in Σ_2^{p}, the second level of the polynomial hierarchy, for $q = 1$, and is in $\mathrm{NP}^{\mathrm{PP}}$ for $q \in (0,1)$. However, in this paper we are interested in P-membership and NP-hardness results only.

4 Plurality and Similar Rules

In this section we focus on q-Cloning for Plurality, Plurality with Runoff, Veto, and Maximin. Surprisingly, these four rules exhibit very similar behavior with respect to cloning.

4.1 Plurality

We start by considering Plurality, which is arguably the simplest voting rule.

Theorem 4.1. *An election is manipulable with respect to Plurality if and only if the manipulator's preferred candidate c does not win, but is ranked first by at least one voter. Moreover, for Plurality 0-cloning can be solved in linear time.*

It is not too hard to strengthen Theorem 4.1 from 0-manipulability to q-manipulability for any $q < 1$.

Theorem 4.2. *For any $q < 1$, a Plurality election is q-manipulable if and only if the manipulator's preferred candidate c does not win, but is ranked first by at least one voter. However, no election is strongly manipulable.*

4.2 Veto and Plurality with Runoff

The Veto rule exhibits extreme vulnerability to cloning.

Theorem 4.3. *Any election is strongly manipulable with respect to Veto. Moreover, for Veto both* 0-CLONING *and* 1-CLONING *can be solved in linear time.*

We now consider Plurality with Runoff. Observe first that cloning any alternative cannot change what happens in the runoff: indeed, if a beats c in their pairwise contest, a would also beat any clone of c in the runoff, and if a loses to c in their pairwise contest, a would also lose in the runoff to any clone of c. Thus, if an alternative c is a *Condorcet loser*, i.e., for any $a \in A \setminus \{c\}$ a strict majority of voters prefers a to c, then c cannot be made a winner by cloning. If it is not a Condorcet loser, then it wins at least one pairwise contest, say against w. Then, if c and w get to the runoff, c would win the election. Further, c and w have a non-zero probability to reach the runoff if both are ranked first at least once. Taken together, these two considerations lead to the following criterion.

Theorem 4.4. *An election is manipulable with respect to Plurality with Runoff if and only if*

(1) the manipulator's preferred candidate c is not a current winner, and

(2) c is not a Condorcet loser and both c and some alternative w that does not beat c in their pairwise election are ranked first by at least one voter each.

Moreover, for Plurality with Runoff 0-CLONING *can be solved in polynomial time.*

As for Plurality, we can characterize q-manipulability for $q \in [0, 1]$. The following theorem can be proved similarly to Theorem 4.2.

Theorem 4.5. *For any $q < 1$, an election is q-manipulable with respect to Plurality with Runoff if and only if it is manipulable with respect to it. However, no election is strongly manipulable.*

4.3 Maximin

Consider the following election that will be used in this section. Let $E = (A, \mathcal{R})$ with $A = \{a_1, \dots, a_k\}$, $\mathcal{R} = (R_1, \dots, R_k)$, where for $i \in [k]$ the preferences of the i-th voter are given by $a_i \succ_i a_{i+1} \succ_i \dots \succ_i a_k \succ_i a_1 \succ_i \dots \succ_i a_{i-1}$. We will refer to any election that can be obtained from E by renaming the candidates as a *k-cyclic election.* In this election, for any $i = 1, \dots, k$, there are $k - 1$ voters that prefer a_{i-1} to a_i (where we assume $a_{k+1} = a_1$). Thus, the Maximin score of each candidate in A is 1. Further, this remains true if we add arbitrary candidates to the election, no matter how the voters rank the additional candidates. This means that, given a candidate $a \in A$, by cloning a and telling the voters to order the clones as in a cyclic election, we can ensure that the Maximin score of any clone of a is 1: in an election with n voters, we create n clones of a and consider the situation where the voters' preferences over those clones form an n-cyclic election. This construction enables us to prove the following result.

Theorem 4.6. *An election is manipulable by cloning with respect to Maximin if and only if the manipulator's preferred candidate c does not win, but is Pareto-optimal. Further, for Maximin* 0-CLONING *can be solved in linear time. No election is strongly manipulable*

It is not clear if one can strengthen the result of Theorem 4.6 to q-manipulability for $0 < q < 1$. This amounts to the following question: suppose that for a fixed n we randomly draw n permutations of $\{1, \ldots, k\}$. Let $P(n,k)$ be the probability that for each $i \in [k]$ there is a $j \in [k]$ such that j precedes i in at least $n-1$ permutations. Is it the case that the probability $P(n,k)$ approaches 1 as $k \to \infty$? Our computations show[3] that this is unlikely to be the case. For $(n,k) = (5,20)$ there was only one success out of 10^6 random trials and only three for $(n,k) = (5,50)$. For both $(n,k) = (7,20)$ and $(n,k) = (7,50)$ not a single random trial out of 10^6 trials was successful. This means that, even if Maximin is q-manipulable for $q > 0$, the number of clones needed would be astronomical.

5 Borda, *k*-Approval, and Copeland

We now consider Borda, k-Approval, and Copeland rules, for which cloning issues get significantly more involved.

5.1 Borda Rule

For Borda rule, just as for Maximin, Pareto-optimality of the manipulator's favorite alternative is necessary and sufficient for the existence of successful manipulation by cloning. However, Borda and Maximin exhibit different behavior with respect to strong manipulability. Moreover, from the point of view of finding an optimal-cost cloning, Borda appears to be harder to deal with than Maximin.

Theorem 5.1. *An election is manipulable by cloning with respect to Borda if and only if the manipulator's preferred candidate c does not win, but is Pareto-optimal. Moreover,* UC 0-CLONING *for Borda can be solved in linear time.*

Briefly, an optimal cloning manipulation for Borda in the UC model is to clone c sufficiently many times and ask all voters to order the clones in the same way. However, for $q > 0$, cloning c is not necessarily optimal.

Strengthening Theorem 5.1 to q-manipulability for some constant q, or to strong manipulability appears to be difficult. We will first characterize the elections that can be strongly manipulated with respect to Borda by cloning the manipulator's favorite candidate.

Proposition 5.2. *An election is strongly manipulable with respect to Borda by cloning the manipulator's preferred candidate c if and only if any candidate whose Borda score is higher than that of c loses to c in a pairwise contest.*

The proof of Proposition 5.2 indicates which orderings of the clones are the most problematic for the manipulator: these are the orderings that, roughly speaking, grant each clone the same number of points. But this is exactly the expected outcome if the orderings are generated uniformly at random! Thus, our proof shows that for Borda, Pareto optimality of the manipulator's most preferred candidate c is insufficient for q-manipulability with $q > 0$ by cloning c only. However, cloning a different candidate may be a better strategy: Suppose that c is Pareto-optimal, and, moreover, the original preference profile contains a candidate c' that is ranked right under c by all voters (one can think of this candidate as an "inferior clone" of c; however, we emphasize that it is present in the original profile). Then one can show that by cloning c' sufficiently many times we can make c a winner with probability 1. However, cloning c itself does not have the same effect if the voters order the clones randomly or adversarially to the manipulator. This is illustrated by the following example.

Example 5.3. Let us consider the following Borda election: $C = \{a, b, c, d\}$, there are four voters v_1, v_2, v_3, v_4, and the preference orders of the voters are:

[3] We are grateful to Danny Chang for his help with these.

$$\begin{array}{ll} v_1 : a \succ c \succ b \succ d & Sc_B(a) = 9 \\ v_2 : a \succ c \succ b \succ d & Sc_B(b) = 4 \\ v_3 : a \succ c \succ b \succ d & Sc_B(c) = 8 \\ v_4 : d \succ c \succ b \succ a & Sc_B(d) = 3 \end{array}$$

The winner here is a with 9 points. However, cloning b into three clones b_1, b_2, b_3 is a 1-manipulation in favor of c since the new score of a is 15 while the new score of c is 16, no matter how clones are ordered. At the same time, no amount of cloning of c can have the same effect. Indeed, after splitting c into $k+1$ clones, the expected score of each clone of c is $4(2 + k/2) = 8 + 2k$, whereas a's score is $9 + 3k$.

This shows that in general, we may need to clone several candidates that are placed between c and its "competitors" in a large number of votes, and determining the right candidates to clone might be difficult. Indeed, it is not clear if a 1-successful manipulation can be found in polynomial time. We thus propose determining the complexity of identifying strongly manipulable profiles with respect to Borda as an open problem.

A related question that is not answered by Theorem 5.1 is the complexity of 0-Cloning in the general cost model. Note that there is a certain similarity between this problem and that of strong manipulability: in both cases, it may be suboptimal to clone c. Indeed, for general costs, we can prove that q-Cloning is NP-hard for any rational q.

Theorem 5.4. *For Borda, q-*Cloning *in the general cost model is* NP*-hard for any $q \in [0, 1]$. Moreover, this is the case even if $p(i, j) \in \{0, 1, \infty\}$ for all $i \in [m], j \in \mathbb{Z}^+$.*

The cost function used in the proof of Theorem 5.4 is very similar to the UC model, except that we are not allowed to clone some of the alternatives.

5.2 k-Approval

Plurality, k-approval and Borda are perhaps the best-known representatives of a large family of voting rules known as *scoring rules*, i.e., rules in which each voter grants each candidate a certain number of points that depends on that candidate's position in the voter's preference order. (Formally, Plurality, k-Approval, and Borda are *families* of scoring rules.) It would be interesting to characterize scoring rules vulnerable to manipulation by cloning. Recall that one can define a scoring rule $\mathcal{F}_\mathbf{w}$ for any vector $\mathbf{w} = (w(1), \dots, w(m))$ with $w(i) \in \mathbb{R}^+ \cup \{0\}$ for $i \in [m]$ (usually, though not always, it is also required that $w(1) \geq \dots \geq w(m)$) as follows: given a preference profile $(R_1, \dots, R_n)$ over a set of alternatives A of size m, the $\mathcal{F}_\mathbf{w}$-score of each alternative $c \in A$ is given by

$$Sc_\mathbf{w}(c) = \sum_{i=1}^{n} w(\mathrm{pos}(c, i)),$$

where $\mathrm{pos}(c, i)$ is the *position* of c in R_i, i.e., $\mathrm{pos}(c, i) = |\{a \in A \mid a \succ_i c\}| + 1$. As usual, the winners are the alternatives with the maximum score. Note, however, that this description does not fit our definition of a voting rule, as it only works for a fixed number of alternatives. To fix this, we will now define scoring rules for infinite rather than finite vectors.

Definition 5.5. *Given a profile $(R_1, \dots, R_n)$ over a set of alternatives A and a monotone sequence $\mathbf{w} = (w(1), \dots,)$, i.e., one that satisfies either (i) $w(1) \leq w(2) \leq \dots$ or (ii) $w(1) \geq w(2) \geq \dots$, we define the $\mathcal{F}_\mathbf{w}$-score of $c \in A$ as $Sc_\mathbf{w}(c) = \sum_{i=1}^{n} w(|A| - \mathrm{pos}(c, i) + 1)$ if $\mathbf{w}$ is non-decreasing and $Sc_\mathbf{w}(c) = \sum_{i=1}^{n} w(\mathrm{pos}(c, i))$ if $\mathbf{w}$ is non-increasing. The winners under $\mathcal{F}_\mathbf{w}$ are the alternatives with the maximum $\mathcal{F}_\mathbf{w}$-score.*

Observe that the Borda rule corresponds to the non-decreasing sequence $(0, 1, 2, 3, \dots)$ and Plurality corresponds to the non-increasing sequence $(1, 0, \dots)$, i.e., we need to consider both non-increasing and non-decreasing sequences to capture well-known scoring rules.

Now, we have observed that even though both Borda and Plurality are susceptible to manipulation by cloning, they exhibit very different behavior with respect to the cloning procedure. Indeed, under Plurality the winner will suffer from cloning, while under Borda her position will usually strengthen (at least as long as we are focusing on manipulability rather than strong manipulability). Further, while no election is strongly manipulable with respect to Plurality, there is a large category of elections that are strongly manipulable with respect to Borda. Thus, an interesting research direction is to determine the relationship between the properties of the sequence $\mathbf{w}$ and the manipulability of the corresponding scoring rule (compare with the work of Hemaspaandra and Hemaspaandra [10] on voter manipulation of scoring rules).

However, this problem is far from being trivial. Indeed, we will now demonstrate that there is a family of scoring rules for which deciding whether a given election is susceptible to cloning is computationally hard. Specifically, this is the case for k-Approval for any $k \geq 2$. We start by showing this for $k = 2$; subsequently, we will generalize our result to the case $k > 2$. Our proof gives a reduction from the problem Dominating Set, defined below.

Definition 5.6. *An instance of the* Dominating Set *problem is a triple (V, E, s), where (V, E) is an undirected graph and s is an integer. We ask if there is a set $W \subseteq V$ such that (a) $|W| \leq s$ and (b) for each $v \in V$ we have $v \in W$ or $(v, w) \in E$ for some $w \in W$.*

Lemma 5.7. *For* 2*-Approval, it is* NP*-hard to decide whether a given election is manipulable by cloning.*

It is not hard to modify the construction in the proof of Lemma 5.7 for the case $k > 2$.

Theorem 5.8. *For any given $k \geq 2$, it is* NP*-hard to decide whether a given election is manipulable by cloning with respect to k-Approval.*

One can also use ideas in the proof of Theorem 5.8 to show that it is NP-hard to decide whether an election is strongly manipulable with respect to k-Approval.

Theorem 5.9. *For any given $k \geq 2$, it is* NP*-hard to decide whether a given election is strongly manipulable by cloning with respect to k-Approval.*

5.3 Copeland

For an election E with a set of candidates A, its *pairwise majority graph* is a directed graph (A, X), where X contains an edge from a to b if more than half of the voters prefer a to b; we say that a *beats* b if $(a, b) \in X$. If exactly half of the voters prefer a to b, we say that a and b are *tied* (this does not mean that their Copeland scores are equal).

For an odd number of voters, the graph (A, X) is a *tournament*, i.e., for each pair $(a, b) \in A^2$, $a \neq b$, we have either $(a, b) \in X$ or $(b, a) \in X$. In this case, we can make use of a well-known tournament solution concept of Uncovered Set [16, 8, 13], defined as follows. Given a tournament (A, X), a candidate a is said to *cover* another candidate b if a beats b as well as every other candidate beaten by b. The *Uncovered Set* of (A, X) is the set of all candidates not covered by other candidates.

It turns out that if the number of voters is odd, the Uncovered Set coincides with the set of candidates that can be made Copeland winners by cloning.

Theorem 5.10. *For any $q \in [0, 1]$, an election E with an odd number of voters is q-manipulable with respect to cloning if and only if the manipulator's preferred candidate c does not win, but is in the Uncovered Set of the pairwise majority graph of E.*

For elections with an even number of voters, the situation is significantly more complicated. The notion of Uncovered Set can be extended to pairwise majority graphs of arbitrary elections in a natural way (see, e.g. [3]): we say that u covers c if u beats c and all alternatives beaten by c, and, in addition, c loses to all alternatives that beat u. In particular, this means that u *does not* cover c if it is beaten by some alternative that is tied with c. This definition generalizes the one for the odd number of voters, However, for an even number of voters, the condition that c is in the Uncovered Set turns out to be necessary, but not sufficient for manipulability by cloning.

Example 5.11. Consider an election with $A = \{a, b, c, u, w\}$. Suppose that a beats u, u beats b, b beats w, w beats a, u and w beat c, and any other pair of candidates is tied. Note that by McGarvey theorem [15] there are voters' preferences that produce this pairwise majority graph. It is easy to see that in this election c cannot be made a winner by cloning even though it is not covered.

Instead, we can characterize cloning-manipulable profiles in terms of the properties of the induced (bipartite) subgraph of (A, X) whose vertices are, on the one hand, the candidates that are tied with c, and, on the other hand, the candidates that beat c as well as all candidates beaten by c. However, it is not clear if this characterization leads to a polynomial-time algorithm. We omit the details due to space constraints.

On the other hand, finding an optimal-cost cloning manipulation is hard even in the UC model.

Theorem 5.12. *For Copeland,* UC q-CLONING *is* NP*-hard for each* $q \in [0, 1]$.

6 Conclusions

We have provided a formal model of manipulating elections by cloning, characterized manipulable and strongly manipulable profiles for many well-known voting rules, and explored the complexity of finding a minimum-cost cloning manipulation. The grouping of voting rules according to their susceptibility to manipulation differs from most standard classifications of voting rules: e.g., scoring rules behave very differently from each other, and Maximin is more similar to Plurality than to Copeland. Future research directions include designing approximation algorithms for the minimum-cost cloning under voting rules for which this problem is known to be NP-hard, and extending our results to other voting rules.

Acknowledgements. Edith Elkind is supported by NRF Research Fellowship (NRF-RF2009-08). Piotr Faliszewski is supported by AGH University of Technology Grant no. 11.11.120.865, by Polish Ministry of Science and Higher Education grant N-N206-378637, and by Foundation for Polish Science's program Homing/Powroty. Arkadii Slinko is supported by the Faculty of Science Research and Development Fund grant 3624495/9844.

References

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

[2] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[3] F. Brandt and F. Fischer. Computational aspects of covering in dominance graphs. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 694–699. AAAI Press, July 2007.

[4] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. AAAI Press, July 2009.

[5] A. Copeland. A "reasonable" social welfare function. Seminar on Mathematics in Social Sciences, University of Michigan, 1951.

[6] E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.

[7] G. Erdélyi, H. Fernau, J. Goldsmith, N. Mattei, D. Raible, and J. Rothe. The complexity of probabilistic lobbying. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory*, pages 86–97. Springer-Verlag *Lecture Notes in Computer Science #5783*, October 2009.

[8] P. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

[9] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[10] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.

[11] G. Laffond, J. Laine, and J. Laslier. Composition consistent tournament solutions and social choice functions. *Social Choice and Welfare*, 13(1):75–93, 1996.

[12] J. Laslier. Rank-based choice correspondencies. *Economics Letters*, 52(3):279–286, 1996.

[13] J. Laslier. *Tournament Solutions and Majority Voting*. Springer-Verlag, 1997.

[14] J. Laslier. Aggregation of preferences with a variable set of alternatives. *Social Choice and Welfare*, 17(2):269–282, 2000.

[15] D. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

[16] N. Miller. Graph theoretical approaches to the theory of voting. *American Journal of Political Science*, 21(4):769–803, 1977.

[17] M. Schulze. A new monotonic and clone-independent single-winner election method. *Voting Matters*, 17:9–19, 2003.

[18] T. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

[19] T. Zavist and T. Tideman. Complete independence of clones in the ranked pairs rule. *Social Choice and Welfare*, 64(2):167–173, 1989.

Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University
Singapore
Email: `eelkind@ntu.edu.sg`

Piotr Faliszewski
Department of Computer Science
AGH University of Science and Technology
Kraków, Poland
Email: `faliszew@agh.edu.pl`

Arkadii Slinko
Department of Mathematics
University of Auckland
Auckland, New Zealand
Email: `slinko@math.auckland.ac.nz`