Manipulating Elections by Changing Voter Perceptions

Junlin Wu, Andrew Estornell, Lecheng Kong and Yevgeniy Vorobeychik
Washington University in St. Louis
{junlin.wu, aestornell, jerry.kong, yvorobeychik}@wustl.edu

Abstract
The integrity of elections is central to democratic systems. However, a myriad of malicious actors aspire to influence election outcomes for financial or political benefit. A common means to such ends is by manipulating perceptions of the voting public about select candidates, for example, through misinformation. We present a formal model of the impact of perception manipulation on election outcomes in the framework of spatial voting theory, in which the preferences of voters over candidates are generated based on their relative distance in the space of issues. We show that controlling elections in this model is, in general, NP-hard, whether issues are binary or real-valued. However, we demonstrate that critical to intractability is the diversity of opinions on issues exhibited by the voting public. When voter views lack diversity, and we can instead group them into a small number of categories—for example, as a result of political polarization—the election control problem can be solved in polynomial time in the number of issues and candidates for arbitrary scoring rules.

1 Introduction
Elections are among the core functional elements of democratic systems. Consequently, there is broad consensus that their integrity is among the top democratic priorities. However, malicious actors may attempt to subvert elections for their own means, whether financial or political [Caldwell et al., 2019; Harper et al., 2019; Khetani-Shah and Deutsch, 2019]. A common approach for manipulating elections is by spreading false information about select candidates, an extreme example of which is the infamous “Pizzagate” campaign targeting Hillary Clinton in the 2016 U.S. presidential election [Robb, 2017]. Less extreme, but far more common, is the spread of misinformation about the positions of candidates on specific issues, such as taxation and debt.

The issue of election vulnerability to malicious manipulation has been studied in the computational social choice literature from a computational complexity perspective under the terms election control (when the election structure itself is manipulated) [Bartholdi et al., 1992; Hemaspaandra et al., 2007; Chen et al., 2017] and bribery (when manipulation is through changing voter preferences over candidates) [Bredereck et al., 2016; Faliszewski and Rothe, 2016].

The traditional study of election control takes voter preferences as given, while considerations of bribery investigate direct manipulations of preference rankings of individual voters. However, neither is a natural model of the impact of misinformation about particular issues on the perceptions of candidates by the voting public. To address this gap, we propose a new model of election control in the spatial voting theory framework. Spatial voting theory explicitly captures voter and candidate positions on issues, with voter preferences over candidates determined by their relative distance in issue space [Anshelevich and Postl, 2016; Anshelevich et al., 2018; Enelow and Hinich, 1984]. In our model of election manipulation, a malicious party can change voter perceptions of a target candidate on issues, subject to a budget constraint (more precisely, we constrain the $l_p$ norm of the manipulation to be below a specified bound). We consider both constructive control, where the malicious goal is to cause the target candidate to win, and destructive control, in which the goal is to cause the target candidate to lose.

We show that when the issues are binary-valued, the problem is hard even with two candidates, for both forms of control and for any $l_p$ norm with integer $1 \leq p < \infty$ used to measure distance in issue space. When issues are real-valued, however, the conclusions for constructive and destructive control differ slightly. For destructive control, the problem is hard even with two candidates. For constructive control, we show hardness for plurality elections when the number of candidates is arbitrary, for $l_p$ norm with integer $1 < p \leq \infty$. However, if there are only two candidates and we measure distance using $l_\infty$ norm, the control problem can be solved in polynomial time. Furthermore, we show that if we restrict either the number of issues or the number of voters to be bounded by a constant, all control problems become tractable, whether issues are binary (for arbitrary $l_p$ norm) or real-valued (for $l_2$ and $l_\infty$), for arbitrary scoring rules used to determine election outcomes. Moreover, we show that the

1This model can also be viewed as an example of bribery, in the sense that the manipulation affects voter preference rankings over candidates. Our use of the term election control is general, referring to any setting in which a malicious party wishes to subvert an election, whatever means they use for doing so.
tractability generalizes even when the number of voters is arbitrary, but their opinions on issues are limited to only a constant (that is, small) number of options.

These seemingly highly technical results offer a broader insight: vulnerability of elections to malicious manipulation of voter perceptions hinges on the extent to which voters exhibit a high diversity of political views. When this is the case, elections are highly resistant to manipulation. However, when voters are Balkanized into a small number of groups with effective uniformity of views within each, for example, due to political polarization, elections become easy to manipulate through misinformation.

Our model of election control is related to several recent studies of election control in the spatial voting theory framework [Lu et al., 2019; Estornell et al., 2020]. However, the means of manipulation in this closely related work is changing the relative importance of issues to voters, whereas our focus is on changing voter perceptions of candidates.

Related Work. The study of election control was initiated by Bartholdi et al. [1992], who studied the impact of adversarially adding, deleting, or partitioning candidates or voters on election outcomes in the constructive control framework. Numerous follow-up efforts extended this analysis in a number of directions, such as destructive control [Hemaspaandra et al., 2007], a variety of voting rules and settings [Menton, 2012; Erdélyi et al., 2015; Chen et al., 2017], as well as when voter preferences can be modified (commonly called the bribery problem) [Bréderèce et al., 2016; Faliszewski and Rothe, 2016] or optimal lobbying [Christian et al., 2007; Binkele-Raible et al., 2014]).

In most election control settings voter preferences are specified directly as preference rankings over the candidates. An alternative approach based on spatial theory of voting, specifies voter and candidate positions on issues, with preference rankings then induced from relative distances between voter and candidate positions [Davis and Hinich, 1968; Enelow and Hinich, 1984; Anshelevich and Postl, 2016; Anshelevich et al., 2018]. Lu et al. [2019] were the first to investigate election control within the spatial theory voting model, with the adversary’s ability restricted to selecting a subset of issues that become the focus of voting preferences. Estornell et al. [2020] study a variation in which an adversary can modify the relative importance of issues in determining voter preferences over candidates. Both are distinct from our model in which the adversary modifies not the importance of issues, but the perceptions of a particular candidate by the voters.

Several models of election control are also motivated by the spread of misinformation about candidates on social networks [Wilder and Vorobeychik, 2018; Castiglioni et al., 2020]. However, these focus on stochastic spread of misinformation in the social influence modeling framework [Kempe et al., 2003], but use the conventional model of elections in which voter preferences are directly specified, with misinformation having a direct impact on a target candidate’s relative ranking for a given voter, rather than an indirect impact stemming from the change in perceived positions on issues, as in our model.

2 Preliminaries

We consider an election with a set of $n$ candidates $C = \{c_1, \ldots, c_n\}$ and $m$ voters $\mathcal{V} = \{v_1, \ldots, v_m\}$. Following spatial voting theory [Enelow and Hinich, 1984], we associate each candidate and voter with a $d$-dimensional vector corresponding to their positions (opinions) on issues, that is, $c_i, v_j \in \mathbb{R}^d$. Each voter $v_j$ ranks candidates in $C$ according to their $l_p$ distance from $v_j$, $\|v_j - c_i\|_p$, with $1 \leq p \leq \infty$ an integer; the closest candidate is ranked 1, and the farthest is ranked $n$ in the list of $v_j$’s preferences. If not mentioned, the parameters of the problem (e.g., $|\mathcal{V}|$, $|C|$ and $d$) are arbitrary. We assume that there are no ties.

In our election control problem, the adversary has a target candidate whose voter perceptions they can manipulate. Without loss of generality, let $c_1$ be the target candidate. We assume that the adversary can change the perception of $c_1$ into $\tilde{c}_1$, subject to the constraint that $\|\tilde{c}_1 - c_1\|_p \leq \epsilon$ for $\epsilon > 0$. This “budget” constraint is natural: for example, if the means for changing perceptions is social media misinformation, the change to perception is likely gradual, and one cannot target arbitrary subsets of issues with an arbitrarily large stream of malicious content. We consider two types of control: constructive, in which the adversary’s goal is for $c_1$ to win the election, and destructive, where the goal is for $c_1$ to lose. While we assume no ties in the actual preference rankings, ties can arise due to adversarial activities; in that case, we always break ties in the adversary’s favor.

We consider election control problems for arbitrary scoring rules. In scoring rules, each candidate $c_i$ ranked $t_{ij}$ by voter $v_j$ receives a score $f(t_{ij})$, where $f : [n] \to \mathbb{R}$ is a non-increasing function. $c_i$ then receives a total score $s_i = \sum_j f(t_{ij})$ from all voters, and the candidate with the highest score $s_i$ wins the election. Many common voting rules are positional, such as plurality ($f(1) = 1$, and $f(t) = 0$ for $t > 1$), veto ($f(n) = 0$ and $f(t) = 1$ for $t \neq n$), Borda ($f(t) = n - t$), and k-approved (for some $1 \leq k \leq n$, $f(t) = 1$ for all $t \leq k$, and $f(t) = 0$ for $t > k$). Note that plurality is a special case of k-approved with $k = 1$.

We study the problem both when issues are binary, i.e., $\mathcal{V} = \{0, 1\}^d$, salient if issues are framed in the form of yes-no questions, such as “do you support leaving the European Union?”, and when issues are real-valued ($\mathcal{V} = \mathbb{R}^d$).

3 Binary-Valued Issues

We begin by studying a special case of our problem in which the issues are binary, that is, $\mathcal{V} = \{0, 1\}^d$, a variant we call Binary Value Perception Manipulation (BVPM).

Definition 3.1 (BVPM). Given a set of candidates $C$, voters $\mathcal{V}$, and $d$ issues, is there a $\tilde{c}_1 \in \mathcal{V} = \{0, 1\}^d$ where $\|\tilde{c}_1 - c_1\|_p \leq \epsilon$ for $\epsilon > 0$ such that $\tilde{c}_1$ wins the election?

Note that for $l_\infty$ the problem is trivial: either $\epsilon \geq 1$, in which case we can set $c_1$ to match any currently winning candidate (and $c_1$ wins by best-case tiebreaking), or $\epsilon < 1$ and we cannot change $c_1$. Thus all the results in this section are for an arbitrary $l_p$ norm for $p$ integer, $1 \leq p < \infty$. Without loss of generality (since the label of 1 or 0 for each issue is arbitrary), we assume that $c_1$ takes a position labeled as 1 for each issue, i.e., $c_1 = [1, \ldots, 1]$. 
We begin by showing that even with 2 candidates and majority voting the BVPM problem is NP-complete. We reduce from Binary Issue Selection Control (BISC), shown by Lu et al. [2019] to be NP-Complete with best-case tie-breaking even when $|c| = 2$.

**Definition 3.2** (BISC [Lu et al., 2019]). Given a set of candidates $C$, voters $V$, and $d$ issues, there is a nonempty subset of binary issues $S \subseteq \{1 : d\}$ such that a target candidate $c_1$ wins the plurality election?

**Theorem 1.** BVPM is NP-complete for constructive and destructive control even with 2 candidates and majority voting.

**Proof.** It is easy to check if a given $c_1$ wins the election; thus, BVPM is in NP. We now show hardness for constructive control by reduction from BISC. Let $\epsilon = (d-1)^{1/p}$, target candidate $c_1 \in \{1, 1, \ldots, 1\}$ and rival candidate $c_2 = \{0, 0, \ldots, 0\}$.

Let the voter set of BVPM be the same as the one in BISC.

Suppose 2-candidate BISC has a solution $S \subseteq \{1 : d\}, S \neq \emptyset$ that will let $c_1$ win the election. For any voter $v_j \in V, j \in [m]$ that votes for target candidate in BISC, by the problem definition of BISC, we have $\sum_{k \in S} |c_1,k - v_j,k| \leq \sum_{k \in S} |c_2,k - v_j,k|^p$. Set $\tilde{c}_1,k = 1$ if $k \in S; \tilde{c}_1,k = 0$ if $k \in S^c$. Since $S \neq \emptyset, |S^c| \leq d - 1, \tilde{c}_1$ is within the budget constraint. For $k \in S$, we have $\sum_{k \in S} |\tilde{c}_1,k - v_j,k|^p = \sum_{k \in S} |c_1,k - v_j,k|^p \leq \sum_{k \in S} |c_2,k - v_j,k|^p$. For $k \in S^c$, since $\tilde{c}_1,k = c_2,k = 0$, we then have $|\tilde{c}_1,k - v_j,k|^p = |c_2,k - v_j,k|^p$. Thus we have $\sum_{k \in S} |\tilde{c}_1,k - v_j,k|^p \leq \sum_{k \in S} |c_2,k - v_j,k|^p$. This means voter $v_j$ votes for $c_1$ in BVPM $(||\tilde{c}_1,v_j|| \leq ||c_2,v_j||)$. $\tilde{c}_1$ wins the election and is a solution to two candidates BVPM.

If two candidates BVPM has a solution $\tilde{c}_1$ that wins the election, by the problem definition of BISC, we have $\sum_{k=1}^d |\tilde{c}_1,k - v_j,k|^p \leq \sum_{k=1}^d |c_2,k - v_j,k|^p$. We let $S = \{k \in [d] \mid \tilde{c}_1,k = 1\}$. Since $\tilde{c}_1,k = c_2,k = 0, k \in S^c$, we have $\sum_{k \in S^c} |\tilde{c}_1,k - v_j,k|^p = \sum_{k \in S} |c_2,k - v_j,k|^p$ since $\tilde{c}_1,k = c_1,k = 1, k \in S$. We then have $\sum_{k \in S} |c_1,k - v_j,k|^p \leq \sum_{k \in S} |c_2,k - v_j,k|^p$. Due to the budget constraint $\epsilon = (d - 1)^{1/p}$, we must have $S \neq \emptyset$, which satisfies the BISC solution requirement. $S$ is a solution to two candidates BISC.

For destructive control, the same argument applies after switching the positions of $c_1$ and $c_2$.

While BVPM is hard in general, we next show that the problem is tractable for a constant number of voters.\footnote{Note that this is trivial for a constant number of binary issues.}

**Theorem 2.** When the number of voters is constant, BVPM can be solved in polynomial time for arbitrary scoring rules, for both constructive and destructive control.

**Proof sketch.** Given an issue $j$, let $v^j$ be a vector corresponding to the position of each voter on issue $j$. The key idea is that when the number of voters $m$ is a constant, there is also a constant number $l \leq 2^m$ of issue equivalence sets, where an equivalence set $I$ is a set of issues with identical $v^j$ (consequently, each issue in $I$ has an identical and interchangeable impact on the election outcome). Since the number of issues of each equivalence set we are allowed to flip is at most $|I| \leq d$, it is direct that we can exhaustively enumerate all $O(d^m)$ possibilities for arbitrary $\epsilon$, which is polynomial since $m$ is constant.

While there is a simple poly-time algorithm for solving BVPM, we can actually considerably improve on its time complexity by leveraging additional problem structure. We begin with the constructive control case for arbitrary scoring rules. A key feature of arbitrary scoring rules is that as long as $c_1$ receives (one of) the highest scores, $c_1$ wins the election. Since the distances between each voter $v_j$ and all candidates $c_i$ other than $c_1$ are fixed, the relative rankings of candidates $c_i (i \geq 2)$ w.r.t. voter $v_j, j \in [m]$ are fixed as well. Given the rankings of candidates $c_i (i \geq 2)$ w.r.t. to voter $v_j$ from closest to furthest as $c_1, \ldots, c_i, \ldots, c_{i-1}$, we can enumerate all scenarios of insertion positions of $c_1$ in this sequence. We denote the final ranking of $c_1$ after insertion w.r.t. voter $v_j$ as $r_j$, meaning $c_1$ will receive a score of $f(r_j)$ from $v_j, j \in [m]$. By going through all scenarios of $c_1$ getting a final ranking position of $(r_1, r_2, \ldots, r_m), r_j \in [m], \forall j \in [m]$, which corresponds to $c_1$ getting a score of $(f(r_1), f(r_2), \ldots, f(r_m))$, we cover all the scenarios of $c_1$ winning. As shown in Lemma 1, this is equivalent to enumerating all scenarios of $c_1$ getting a ranking position higher than $r_j \in [m]$ w.r.t. $v_j$ for $j \in [m]$. The missing proofs of this and other results are provided in the Supplement.

**Lemma 1.** For constructive control, if a ranking position $(r_1, r_2, \ldots, r_m)$ is feasible for $c_1$ under the budget constraint and lets $c_1$ win the election, then for a ranking position $(r_1', r_2', \ldots, r_m')$ that is feasible with $r_j' \leq r_j, \forall j \in [m]$, it will also let $c_1$ win the election.

This means by enumerating all scenarios of $c_1$ getting a ranking position higher than $(r_1, r_2, \ldots, r_m)$, which corresponds to $c_1$ getting a score of at least $(f(r_1), f(r_2), \ldots, f(r_m))$, we cover all the possible scenarios of $c_1$ winning. In fact, we can further simplify the calculation by only enumerating some ranking positions each corresponding to a unique $f$ score value. Given an arbitrary scoring function $f$ that has $r$ unique values $(\{f_{\text{value}} = r\})$, since $f$ is non-increasing, we can partition the domain of $f$ by $0 = s_0 < s_1 < \cdots < s_r = n$, so that $(f(k))_{k=s_i+1}^{s_{i+1}}$ have the same value, $i \in \{0, 1, \ldots, r - 1\}$. This means $s_{i+1}$ is the lowest ranking position that corresponds to score $f(s_{i+1})$ and $\{f(s_i)\}_{i=1}^{r}$ contains all the unique values of $f$.

**Lemma 2.** For constructive control, by enumerating all scenarios of $c_1$ getting a ranking position higher than $t_j$ w.r.t. voter $v_j$ for $t_j \in \{s_1, s_2, \ldots, s_r\}, \forall j \in [m]$, we cover all the possible scenarios of $c_1$ winning.

Next we solve the problem for each $(t_1, \ldots, t_m)$ scenario with $t_j \in \{s_1, s_2, \ldots, s_r\}, \forall j \in [m]$. For each voter $v_j, j \in [m]$, we rank candidates $c_i (i \geq 2)$ by their distances to voter $v_j$ from closest to furthest, and use $d_i'$ to denote the distance between $v_j$ and the candidate ranked $t_j$-th closest to it. Since
the tie breaks in the adversary’s favor, as long as \( c_1 \)’s distance to \( v_j \) is no more than \( d_{ij}^1 \), \( c_1 \) will receive a score of at least \( f(t_j) \) from \( v_j \). Notice that since the rankings of \( c_i (i \geq 2) \) do not include \( c_1 \), only \( d_{ij}^0 \) for \( 1 \leq t_j \leq n-1 \) are properly defined. For \( t_j = s_j = n \), we let \( d_{ij}^n = +\infty \), since \( c_1 \) is guaranteed to get at least the lowest score \( f(n) \).

For each scenario, the problem can be represented as the following integer linear constraint problem:

\[
x_i \leq \min\{b_i, [e^p] \}, \quad 1 \leq i \leq 2^m \tag{1a}
\]

\[
\sum_{i=1}^{2^m} z_{ij} x_i + (d_{ij}^0)^p \leq (d_{ij}^1)^p, \quad 1 \leq j \leq m \tag{1b}
\]

\[
\sum_{i=1}^{2^m} x_i \leq [e^p], \quad x_i \in \mathbb{Z}^+, \tag{1c}
\]

where \( x_i \) is the number of issues in an issue equivalence class \( i \) that we want to flip to 0, \( b_i \) is the size of the \( i \)-th issue equivalence class (where \( \sum_i b_i = d \), \( d_{ij}^0 \) is the original distance between target candidate \( c_1 \) and voter \( v_j \), \( z_{ij} \in \{-1, +1\} \) is the sign of impact of flipping the issue: \( z_{ij} = +1 \) if previously the \( j \)-th voter in any issue in the \( i \)-th equivalence class is 1 (since flipping the \( c_1 \) to 0 will increase the distance) and \( z_{ij} = -1 \) if previously it is 0, since flip the target candidate to 0 will decrease the distance. Since the size of the input of this integer feasibility problem is \( O(\log(d)) \) and the number of variables is constant, it can be solved in time \( O(\log(d)) \) [Lokshtanov, 2009, Theorem 2.8.1]

The total number of times we need to run the ILP and check whether \( c_1 \) wins the election is bounded by the number of unique \((t_1, \ldots, t_m)\) scenarios, which is \( [f_{\text{uniq}}]^m \). For an arbitrary scoring function \( f \), the time complexity for calculating voter-candidate distances is \( O(nd) \), ranking distances takes \( O(n \log(n)) \) time. The calculation of the issue equivalence sets takes \( O(d) \) time. For each scenario, solving the ILP takes \( O(\log(d)) \) time; calculating the distance between \( c_1 \) and all voters takes \( O(d) \) time; checking whether \( c_1 \) wins the election takes \( O(n) \) time. The total time complexity of the algorithm is \( O(n(d + \log(n)) + [f_{\text{uniq}}]m(d + n)) \).

If, in addition, \( |f_{\text{uniq}}| = r \) is constant (e.g., for plurality), then the number of scenarios \( |f_{\text{uniq}}|^m \) is constant. Moreover, we do not need to do a total sort for the distances. Through finding the \( t_j \)-th order statistics and Quicksort Partition, the total time complexity of the algorithm is \( O(nd) \).

In Supplement B we present a similar analysis and algorithm as above for destructive control. In either case, the complexity is linear in the dimension \( d \) of the issue space.

As noted earlier, considering a constant number of voters may seem unrealistic. However, we note that these algorithms are straightforward to generalize to a setting with an arbitrary number of voters, but in which the positions of voters on issues, \( v_j \), can only take on values from a small collection of possibilities \( Q \) (that is, \(|Q| \) is a constant, and for each voter \( j \), \( v_j \in Q \)). Specifically, the only change is to calculate the weighted final score in each case above, where the weight for each distinct voter position (opinion) type \( q \in Q \) is the number of voters \( y \) with position \( v_j = q \). This is expressed in the following corollary.

**Corollary 1.** RVPM can be solved in polynomial time when the number of distinct voter opinions is constant for constructive and destructive control for arbitrary scoring rules.

The key insight that our results offer is that the complexity of manipulating elections by changing voter perceptions about candidates hinges on the diversity of opinions among voters. In particular, when voters hold a broad diversity of views, manipulation is intractable; if, in contrast, voters are siloed into a relatively small collection of echo chambers, subverting elections becomes easy. Below, we show that this observation extends to real-valued issues.

## 4 Real-Valued Issues

Next, we turn to Real Value Perception Manipulation (RVPM), the problem identical to BVPM except that now the issue space \( I \) is real-valued.

### 4.1 Hardness Results

We begin by showing that nearly every variant of RVPM is, in general, computationally intractable. First, we show that the election control problem is hard under \( l_p \) norm with integer \( 1 < p \leq \infty \) under destructive control even with 2 candidates and majority voting, and constructive control even for plurality voting. Nonetheless, we show that constructive control with \( l_\infty \) norm and only two candidates is in \( P \).

Our hardness result for destructive control uses a reduction from 3-SAT, the proofs are deferred to the supplement (A.3 and A.4).

**Theorem 3.** The destructive control variant of RVPM is \( NP \)-complete under \( l_p \) norm for integer \( 1 < p \leq \infty \) even with two candidates and majority voting.

The following theorem (proved in Supplements A.5 and A.6) shows that constructive control is also hard.

**Theorem 4.** The constructive control variant of RVPM under \( l_p \) norm for integer \( 1 < p \leq \infty \) is \( NP \)-complete for plurality voting.

Note that Theorem 4 is stated for an arbitrary number of candidates. For two candidates and \( l_\infty \) norm, however, constructive variant of RVPM is easy.

**Theorem 5.** The constructive control \( l_\infty \) norm variant of RVPM with 2 candidates can be solved in time \( O(nd) \) for arbitrary scoring rules.

The proof is provided in Supplement A.7. Note that our results do not resolve the question of constructive control with \( p < \infty \); we leave it as an open question.

Next, we consider two restrictions of RVPM: 1) assuming a constant number of issues, and 2) assuming a constant number of distinct voter opinions. In all these restricted cases, we show how to solve RVPM in polynomial time for \( l_2 \) and \( l_\infty \) norm. We leave the problem open for arbitrary \( l_p \) norms.

### 4.2 Constant Number of Issues

When the number of issues is constant, we show that RVPM is tractable for \( l_2 \) and \( l_\infty \) norms (our focus on these two norms follows the precedent from prior literature [Crama et al., 1995; Crama and Ibaraki, 1997]). Note that unlike with
binary issues, tractability of RVPM with a constant number
of issues is non-trivial since the issue space is continuous and
cannot be exhaustively searched in finite time.

Constructive Control
We start studying the problem of constructive control, with
RVPM in that case closely related to the well-known product
positioning and ball intersection problems [Crama et al.,
1995]. The goal in product positioning is to find \( x \in \mathbb{R}^m \) that
maximizes the number of consumers for whom \( x \) is closer
(in \( l_p \)) to their ideal product than any of the competitors. The
ball intersection problem aims to maximize the weighted sum
of \( l_p \) balls to which \( x \) belongs. Neither exactly captures our
problem given the presence of the attacker budget constraint
and different scoring scenarios, but both are useful tools in
constructing the algorithms for our problem below.

Theorem 6. RVPM can be solved in polynomial time under
\( l_2 \) norm when the number of issues is constant for constructive
control for arbitrary scoring rules.

Proof. We first convert our problem into the ball intersection
problem. Let \( B_j^{t_j} \) be the ball that corresponds to voter \( v_j \),
with radius \( d_j^{t_j} \) as defined in the constructive control variant
of BVPM, that is,
\[
B_j^{t_j} = \{ \hat{c}_1 \in \mathbb{R}^d | \|\hat{c}_1 - v_j\|_2 \leq d_j^{t_j} \}.
\]
Similarly, we define the candidate budget ball
\[
B_c = \{ \hat{c}_1 \in \mathbb{R}^d | \|\hat{c}_1 - c_1\|_2 \leq \epsilon \}.
\]
Since the tie breaks in the adversary’s favor, \( \hat{c}_1 \) will receive
a score of at least \( f(t_j) \) from voter \( v_j \) iff \( \hat{c}_1 \) falls within \( B_j^{t_j} \)
and \( B_c \). According to Lemma 2 (which also applies
when issues are real-valued), by finding a representative point \( \hat{c}_1 \)
within \( \{ B_c \} \cup (\bigcup_{j=1}^m \{ B_j^{t_j} \}) \) for all scenarios of \( t_1, \ldots, t_m \)
with \( t_j \in \{ s_1, \ldots, s_r \} \) (partition of the domain of \( f \) based on
unique values as defined in the constructive control variant
of BVPM), \( \forall j \in [m] \), we cover all the scenarios of \( c_1 \) winning.
We can now directly apply the ball intersection algorithm by
Crama et al. [1995] for \( l_2 \) and constant \( d \) to our problem only
once for \( \{ B_c \} \cup (\bigcup_{j=1}^m \bigcup_{i=1}^k \{ B_j^{t_j} \}) \), the resulting set \( P \) (representative points of
intersections) contains a representative point for all the scenarios.
We check for each point in \( P \) whether it is within \( B_c \) and
wins the election. The time complexity of the algorithm is exponential
only in \( d \).

A similar result, based on a similar connection to box intersection,
can be obtained for the \( l_{\infty} \) norm.

Theorem 7. RVPM can be solved in polynomial time under
\( l_{\infty} \) norm when the number of issues is constant for constructive
control for arbitrary scoring rules.

Destructive Control
Next we study the problem under destructive control. Define
an open voter ball corresponding to voter \( v_j \) with radius \( d_j^{t_j} \)
under \( l_2 \) norm as \( B_j^{t_j} = \{ \hat{c}_1 \in \mathbb{R}^d | \|\hat{c}_1 - v_j\|_2 < d_j^{t_j} \} \), and
recall that \( d_j^{t_j} \) is the distance between \( v_j \) and the candidate
ranked \( t_j \)-th closest to it. The next lemma, proved in the Supple-
ment, provides an important building block.

Lemma 3. Given \( k \) open balls \( \{ B_1, \ldots, B_k \} \) and a closed
ball \( B_c \), let \( P \) be the representative points of intersections (de-
cined in Crama et al. [1995]) w.r.t. \( \{ B_c, B_1, \ldots, B_k \} \), where
\( B_j \) is the closed ball corresponds to \( B_j, \forall j \in [k] \). For any
given family of open balls \( \{ B_{i_1}, \ldots, B_{i_t} \} \subseteq \{ B_1, \ldots, B_k \} \),
let \( P' = \{ x | x \notin B_j, \forall j \in \{ i_1, \ldots, i_t \}, x \in B_c \} \). If \( P' \neq \emptyset \),
then \( P \cap P' \neq \emptyset \).

Next, we show that when the number of issues is constant,
the destructive control variant of RVPM is tractable.

Theorem 8. When the number of issues is constant, the de-
structive control variant of RVPM can be solved in polynomial
time under \( l_2 \) norm for arbitrary scoring rules.

Proof. Since the tie breaks in the adversary’s favor, within
\( B_c, \hat{c}_1 \) will get a score of no more than \( f(t_j) \) from voter \( v_j \) iff
it falls outside of \( B_j^{t_j} \). Similar to Theorem 6, we cover all the
scenarios of \( c_1 \) losing by finding the set that contains a repre-
sentative point within \( B_c \) that falls outside of \( \bigcup_{j=1}^m \{ B_j^{t_j} \} \) for
all scenarios of \( t_1, \ldots, t_m \) with \( t_j \in \{ s_1, \ldots, s_r \} \) (partition of
the domain of \( f \) based on unique values as defined in the
destructive control variant of BVPM), \( \forall j \in [m] \). Lemma 3
shows us that the set \( P \) (representative points of intersections)
for the family of balls \( \{ B_c \} \cup (\bigcup_{j=1}^m \bigcup_{i=1}^k \{ B_j^{t_j} \}) \) contains
a representative point for all the scenarios. The problem could be
solved in polynomial time with minor modifications to the
algorithm in Theorem 6.

The destructive control problem with \( l_{\infty} \) norm involves
solving a non-convex feasibility problem. The next lemma
shows that the problem of relevance can nevertheless be
solved in polynomial time.

Lemma 4. For the feasibility problem
\[
\begin{align*}
\|\hat{y} - y\|_{\infty} & \leq \epsilon \\
\|\hat{y} - a_i\|_{\infty} & \geq b_i, \quad i \in [k] \\
\epsilon & > 0, b_i > 0, \hat{y}, y, a_i \in \mathbb{R}^d,
\end{align*}
\]
if \( \hat{y} \in \mathbb{R}^d \) satisfy constraint (2a) and \( \bigcup_{j=1}^d S_j(\hat{y}_j) = [k] \),
where \( S_j(\hat{y}_j) = \{ i \in [k] | \|\hat{y} - a_i\|_{\infty} \geq b_i \} \), \( \forall j \in [d] \),
then \( \hat{y} \) is a solution to the feasibility problem. Moreover, for
all \( j \in [d] \), let set \( P_j = \bigcup_{z=1}^{b_i} \{ (-b_i + a_{i,j}, b_i + a_{i,j}) \cap
[-\epsilon + y_j, \epsilon + y_j] \} \), or \( P_j = \{ y_j \} \) if the set is empty, then
\( P = \{ p \in \mathbb{R}^d | p_j \in P_j, \forall j \in [d] \} \) contains
a representative solution point \( \hat{y} \) to the problem.

We use Lemma 4 to show that the destructive control vari-
ant of RVPM is also tractable for the \( l_{\infty} \) norm.

Theorem 9. RVPM can be solved in polynomial time under
\( l_{\infty} \) norm when the number of issues is constant for destructive
control for arbitrary scoring rules.

Proof. We solve the problem by using Lemma 4 to find the
set \( P \) which contains a representative solution point for all
the scenarios of \( t_1, \ldots, t_m \) (defined as in Theorem 8). Each
scenario represents finding a \( \hat{c}_1 \) within the budget constraint
that gets a score of no more than \( f(t_j) \) from voter \( v_j, j \in [m] \):
\[
\begin{align*}
\|\hat{c}_1 - c_1\|_{\infty} & \leq \epsilon \\
\|\hat{c}_1 - v_j\|_{\infty} & \geq d_j^{t_j}, \quad j \in [m]
\end{align*}
\]
where \(d_j\) is defined in the destructive control variant of BVPM. Notice that there are in total \(m \cdot \lfloor f_{\text{uniq}} \rfloor\) open hypercubes involved. According to Lemma 4, the representative solution set \(P\) for all scenarios has at most \(2m \cdot \lfloor f_{\text{uniq}} \rfloor\) choices for each dimension. The time complexity of the algorithm is exponential only in \(d\).

\[\square\]

4.3 Constant Number of Distinct Voters

Next, we turn to the case when the number of distinct voter opinion vectors \(v_j\) is bounded by a constant. As in Section 3, we simplify the discussion by assuming that the number of voters is constant; generalization to an arbitrary number of voters whose opinions can be grouped into a small set \(Q\) of possibilities is straightforward using the same idea as for BVPM. For \(l_2\) norm, we use the ball intersection algorithm as in Section 4.2.

**Theorem 10.** When the number of voters is constant, the constructive and destructive control variants of RVPM can be solved in polynomial time under the \(l_2\) norm for arbitrary scoring rules.

**Proof.** For constructive control, we solve the problem for each scenario of \((t_1, \ldots, t_m)\) (defined as in Theorem 6) through finding a representative point within \(\{B_i\} \cup (\bigcup_{j=1}^{m} B_j^{t_j})\), which takes \(O(d^3)\) time using a variation of the ball intersection algorithm by Crama et al. [1995] as in Theorem 6. Since there are in total \(O(\lfloor f_{\text{uniq}} \rfloor)\) scenarios, the problem can be solved in polynomial time. For destructive control, a similar argument holds and the problem can be solved by using a variation of the ball intersection algorithm as in Theorem 8. \[\square\]

For \(l_\infty\) norm, the constructive case can be solved by the application of linear programming.

**Theorem 11.** When the number of voters is constant, the constructive control variant of RVPM can be solved in polynomial time under the \(l_\infty\) norm for arbitrary scoring rules.

**Proof.** For each scenario of \((t_1, \ldots, t_m)\) (defined as in Theorem 6), we solve the below linear programming:

\[
\begin{align*}
||\hat{c}_1 - c_1||_\infty &\leq \epsilon \quad (4a) \\
||\hat{c}_1 - v_j||_\infty &\leq d_j^t, \quad j \in [m] \quad (4b)
\end{align*}
\]

Since each LP has \(O(d)\) linear constraints, and there are \(O(\lfloor f_{\text{uniq}} \rfloor)\) scenarios, the problem is polynomial time solvable. Alternatively, we can check the interval endpoints similar to Lemma 4 for each dimension. Since \(m\) is a constant, the algorithm returns \(\hat{c}_1\) in \(O(d)\) time if a solution to the LP exists, or NO if not feasible.

\[\square\]

The destructive case is somewhat more involved. We begin with a lemma that again shows that a non-convex feasibility problem we need to solve is tractable.

**Lemma 5.** The feasibility problem in Lemma 4 can be solved in linear time if the number of constraints \(k\) is constant.

**Proof sketch.** According to Lemma 4, for each dimension \(j \in [d]\), we could compute set \(P_j\) and its corresponding set \(S_j = \{S_j(p_j) | p_j \in P_j\}\). Finding a solution to the feasibility problem is equivalent to finding \(S_j(p_j) = S_j\), for all \(j \in [l]\), with some \(l \leq d\) that satisfy \(\cup_{j=1}^{l} S_j(p_j) = [k]\), and \([p_1, p_2, \ldots, p_l, y_{l+1}, y_{l+2}, \ldots, k]\) is a solution. Since \(k\) is constant, the algorithm is linear and of complexity \(O(d)\). We can also determine in linear time if none of the representative points satisfy the feasibility condition, and return NO in that case. The detailed algorithm and proof are provided in Supplement C. \[\square\]

**Theorem 12.** When the number of voters is constant, the destructive control variant of RVPM can be solved in polynomial time under \(l_\infty\) norm for arbitrary scoring rules.

**Proof.** Using Lemma 5, we can solve the linear feasibility problem in Equation (3) where \(m\) is constant for each scenario of \((t_1, \ldots, t_m)\) (defined as in Theorem 8). Each feasibility problem can be solved in \(O(d)\) time and there are \(O(\lfloor f_{\text{uniq}} \rfloor)\) scenarios. The full time complexity of the algorithm is \(O(n(d + \log(n))) + \lfloor f_{\text{uniq}} \rfloor m + n(d + n)\), or \(O(nd)\) if \(\lfloor f_{\text{uniq}} \rfloor\) is constant.

We can extend the results to an arbitrary number of voters with a constant number of distinct opinions by the same argument as for BVPM.

5 Conclusion

We model the impact of political misinformation on elections as election control in the spatial model of voting in which an adversary manipulates perceptions of the positions of a target candidate by the voters. Our central observation, which obtains both when issues are real-valued and binary, and for different ways we can measure distance in generating preferences over candidates based on their relative positions to voters, is that what matters is the extent of opinion diversity in the voting population. Specifically, when voter positions on issues are highly diverse, the manipulation problem is tractable in most settings. In contrast, when voter views can be reduced by a small number of opinion groups, the control problem becomes linear in dimension when issues are binary, and polynomial with real-valued issues. Our characterization of the complexity landscape leaves several open questions, such as hardness of constructive control with two candidates in the setting with real-valued issues (we only show that it is tractable for \(l_\infty\) norm). Furthermore, our negative results for real-valued issues do not address the case of \(l_1\), while our positive results only apply to \(l_2\) and \(l_\infty\) in this setting.

Our model has several important limitations that suggest further useful future directions. First, we assume that the same norm is used both by voters to rank candidates, and to limit the extent of perception manipulation; however, these distances may often be useful to measure in different ways. Second, we assume that perception manipulation has identical impact on all voters. A more sophisticated model would blend this with the election control approaches in which misinformation spreads through a social network, with only a subset of voters impacted, potentially in different ways.
Acknowledgements This work was partially supported by the National Science Foundation (grants IIS-1905558, IIS-1903207, and IIS-1939677) and Amazon.

References

[Anshelevich and Postl, 2016] Elliot Anshelevich and John Postl. Randomized social choice functions under metric preferences. In International Joint Conference on Artificial Intelligence, page 46–52, 2016.

[Anshelevich et al., 2018] Elliot Anshelevich, Onkar Bhardwaj, Edith Elkind, John Postl, and Piotr Skowron. Approximating optimal social choice under metric preferences. Artificial Intelligence, 264:27–51, 2018.

[Bartholdi et al., 1992] John J. Bartholdi, Craig A. Tovey, and Michael A. Trick. How hard is it to control an election? Mathematical and Computer Modelling, 16(8):27–40, 1992.

[Binkele-Raible et al., 2014] Daniel Binkele-Raible, Gábor Erdélyi, Henning Fernau, Judy Goldsmith, Nicholas Mattiei, and Jörg Rothe. The complexity of probabilistic lobbying. Discrete Optimization, 11:1–21, 2014.

[Bredereck et al., 2016] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Large-scale election campaigns: Combinatorial shift bribery. Journal of Artificial Intelligence Research, 55:603–652, 2016.

[Caldwell et al., 2019] L. A. Caldwell, H. Przybyla, and K. Stewart. Senate intelligence report finds ‘extensive’ Russian election interference. In NBC News. 2019.

[Castiglioni et al., 2020] Matteo Castiglioni, Diodato Ferraioli, and Nicola Gatti. Election control in social networks via edge addition or removal. In AAAI Conference on Artificial Intelligence, pages 1878–1885. 2020.

[Chen et al., 2017] Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Elections with few voters: Candidate control can be easy. Journal of Artificial Intelligence Research, 60(1):937–1002, 2017.

[Christian et al., 2007] Robin Christian, Mike Fellows, Frances Rosamond, and Arkadii Slinko. On complexity of lobbying in multiple referenda. Review of Economic Design, 11(3):217–224, 2007.

[Crama and Ibaraki, 1997] Yves Crama and Toshiohide Ibaraki. Hitting or avoiding balls in euclidean space. Annals of Operations Research, 69:47–64, 1997.

[Crama et al., 1995] Yves Crama, Pierre Hansen, and Brigitte Jaumard. Complexity of product positioning and ball intersection problems. Mathematics of Operations Research, 20(4):885–894, 1995.

[Davis and Hinich, 1968] Otto A. Davis and Melvin J. Hinich. On the power and importance of the mean preference in a mathematical model of democratic choice. Public Choice, 5(1):59–72, 1968.

[Enelow and Hinich, 1984] James M. Enelow and Melvin J. Hinich. The Spatial Theory of Voting: An Introduction. Cambridge University Press, 1984.

[Erdélyi et al., 2015] Gábor Erdélyi, Michael R. Fellows, Jörg Rothe, and Lena Schend. Control complexity in bucklin and fallback voting: A theoretical analysis. Journal of Computer and System Sciences, 81(4):632–660, 2015.

[Estornell et al., 2020] Andrew Estornell, Sanmay Das, Edith Elkind, and Yevgeniy Vorobeychik. Election control by manipulating issue significance. In Conference on Uncertainty in Artificial Intelligence, pages 340–349, 2020.

[Faliszewski and Rothe, 2016] Piotr Faliszewski and Jörg Rothe. Control and bribery in voting. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, Handbook of Computational Social Choice, pages 146–168. Cambridge University Press, 2016.

[Harper et al., 2019] T. Harper, C. Wheeler, and R. Kerbaj. Revealed: the Russia report. In The Sunday Times. 2019.

[Hemaspaandra et al., 2007] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Anyone but him: The complexity of precluding an alternative. Artificial Intelligence, 171:255–285, 2007.

[Imai and Asano, 1983] Hiroshi Imai and Takao Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. Journal of Algorithms, 4(4):310–323, 1983.

[Kempe et al., 2003] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 137–146, 2003.

[Khetani-Shah and Deutsch, 2019] S. Khetani-Shah and J. Deutsch. Brexit timeline: From referendum to EU exit. In Politico Pro. 2019.

[Lee, 1983] Der-Tsai Lee. Maximum clique problem of rectangle graphs. Advances in Computing Research, 1:91–107, 1983.

[Lokshtanov, 2009] Daniel Lokshtanov. New methods in parameterized algorithms and complexity. PhD thesis, University of Bergen, Norway, 2009.

[Lu et al., 2019] Jasper Lu, David Kai Zhang, Zinovi Rabinovich, Svetlana Obraztsova, and Yevgeniy Vorobeychik. Manipulating elections by selecting issues. In International Conference on Autonomous Agents and Multiagent Systems, page 529–537, 2019.

[Megiddo, 1990] Nimrod Megiddo. On the complexity of some geometric problems in unbounded dimension. Journal of Symbolic Computation, 10(3-4):327–334, 1990.

[Menton, 2012] Curtis Menton. Normalized range voting broadly resists control. Theory of Computing Systems, 53(4):507–531, 2012.

[Robb, 2017] Amanda Robb. Anatomy of a fake news scandal. In Rolling Stone. 2017.

[Wilders and Vorobeychik, 2018] Bryan Wilder and Yevgeniy Vorobeychik. Controlling elections through social influence. In International Conference on Autonomous Agents and Multiagent Systems, page 265–273, 2018.
Supplementary Materials

A  Missing Proofs

A.1 Proof of Lemma 1

For each voter $v_j, j \in [m]$, we rank candidates $c_i (i \geq 2)$ by their distances to $v_j$ from closest to furthest. Without loss of generality, we assume candidates $c_2, c_3, \ldots, c_n$ are ranked from closest to furthest w.r.t. $v_j$. Then when $c_1$ is inserted into this sequence and is ranked $r_j$, the score each candidate receives from $v_j$ is as below:

$$
\begin{array}{cccccc}
   c_2 & \ldots & c_{r_j} & c_1 & c_{r_j+1} & \ldots & c_n \\
   f(1) & \ldots & f(r_j-1) & f(r_j) & f(r_j+1) & \ldots & f(n) \\
\end{array}
$$

If we move $c_1$ from ranking position $r_j$ to $r_j'$, $r_j' \leq r_j$, since $f$ is a non-increasing function, the score $c_1$ receives from $v_j$ will increase from $f(r_j)$ to $f(r_j')$; the scores candidates $c_{r_j+1}, c_{r_j+2}, \ldots, c_n$ and $c_2, c_3, \ldots, c_{r_j}$ receive from $v_j$ will not change; for candidates $c_{r_j+1}, c_{r_j+2}, \ldots, c_n$, since their rankings will decrease by 1, the scores they receive from $v_j$ will not increase. This means if we move $c_1$ from ranking position $(r_1, r_2, \ldots, r_m)$ to $(r_1', r_2', \ldots, r_m')$, $r_j' \leq r_j, \forall j \in [m]$, the total score $c_1$ receives will not decrease, while the total scores other candidates receive will not increase. Since $c_1$ wins the election with ranking position $(r_1', r_2', \ldots, r_m)$, $c_1$ will also win the election with ranking position $(r_1', r_2', \ldots, r_m')$.

A.2 Proof of Lemma 2

Without loss of generality, we assume candidates $c_2, c_3, \ldots, c_n$ are ranked from closest to furthest w.r.t. voter $v_j$. We next insert $c_1$ into this sequence and assume $c_1$ receives a score of $f(s_{i+1})$ from $v_j$, where $i \in \{0, 1, \ldots, r-1\}$. After the insertion, the score each candidate receives from $v_j$ is as below:

$$
\begin{array}{cccccc}
   \ldots, c_{s_{i+1}}+1 & \ldots, c_{s_{i+1}}+1, \ldots, c_{s_{i+2}} & c_{s_{i+2}+1, \ldots, c_{s_{i+3}}+1} & \ldots, c\n
   f(s_i) & f(s_{i+1}) & f(s_{i+2}) & f(s_{i+3}) \\
\end{array}
$$

Notice that whichever ranking position $c_1$ takes among $s_i+1, s_i+2, \ldots, s_{i+1}$, the final score each candidate receives from $v_j$ are the same, meaning these ranking positions are equivalent. Since the solution set of getting a ranking position higher than $s_i+1$ covers the solution set of getting a ranking position higher than $s_i+1, s_i+2, \ldots, s_{i+1}-1$, we only need to consider $s_i+1$.

A.3 Proof of Theorem 3 ($l_p$ norm, $1 < p < \infty$)

Checking whether target candidate $c_i$ loses the election is clearly in P. We next prove hardness by reduction from 3-SAT. The proof is in $\mathbb{R}^{d+1}$ space.

Definition A.1 (3-SAT). Given a set of variables $X$, a set of clauses $C$ over $X$ where $\forall c \in C$ has $|c| = 3$, is there an assignment to the variables $X$ that satisfies all the clauses?

Let target candidate $c_1 = [0, 0, \ldots, 0, 0]$; rival candidate $c_2 = [0, 0, \ldots, 0, a]$ with $a = (\frac{d}{d-1})^p + (\frac{1}{d-1})^p(d-1) - 1$, and $\epsilon = d^{\frac{1}{d-1}}$. Given a 3-SAT instance with $d-1$ variables $\{X_1, \ldots, X_{d-1}\}$ and $r$ clauses $\{E_1, E_2, \ldots, E_r\}$, we create a voter profile $v_D = [0, 0, \ldots, 0, -a]$ which is adopted by $r$ voters. Note that since the distance between $c_2$ and $v_D$ is $2a$, while the maximum distance between $c_i$ and $v_D$ is $a + d^{\frac{1}{d-1}} < 2a$ (max $\|c_i - v_D\|_p$, s.t. $\|\tilde{c}_i\|_p \leq c$), whenever $c_i$ moves, $v_D$ will always vote for $c_1$. Let $e_i \in \mathbb{R}^{d+1}$ be the unit vector that has 1 in the $i$-th position. We create 2d voters $\{v_{e_i} \in v_{-e_i}, i = 1\}$ where $v_{e_i} = e_i, v_{-e_i} = -e_i, i \in [d]$. Next we create $r$ voters $\{v_i\}_{i=1}^r$ where each $v_i$ corresponds to clause $E_i, i \in [r]$. Given fixed parameters $\alpha > 0, l > 0$ (which we will specify later), for each $v_i, v_{i,j} = -\alpha$ if $X_j$ is included in clause $E_i, v_{i,j} = \alpha$ if $X_j$ is included in clause $E_i, j \in [d-1]$; $v_i,d = \alpha$; otherwise.

We now specify parameters $\alpha$ and $l$. We chose $\alpha > 0$ and $l > 0$ so that they satisfy the below inequality:

$$
\begin{align*}
(\frac{1}{d} + \alpha)^p &+ 2(\frac{1}{d} - \alpha)^p + \frac{1}{d}\left((\frac{1}{d} - \alpha)^p + (\frac{1}{d} + \alpha)^p\right) > a^p + 3a^p + (l\alpha)^p \\
&> 3\left(\frac{1}{d} - \alpha\right)^p + \frac{1}{d}\left((\frac{1}{d} - \alpha)^p + (\frac{1}{d} + l\alpha)^p\right)
\end{align*}
$$

We claim such parameters must exist. We first move term $3a^p + (l\alpha)^p$ to the right and left side of the inequality. Since $d$ and $p$ are given, $a^p$ is a fixed value in $\mathbb{R}$, and the right and left side of the inequality are continuous w.r.t. $\alpha$ and $l$, with gap equals to $(\frac{1}{d} + \alpha)^p - (\frac{1}{d} - \alpha)^p > 0$. This means if we could find (a) $l > 0$, $\alpha > 0$ that satisfy $(\frac{1}{d} + \alpha)^p + 2(\frac{1}{d} - \alpha)^p + \frac{1}{d}\left((\frac{1}{d} - \alpha)^p + (\frac{1}{d} + \alpha)^p\right) > a^p + \alpha^p$ and (b) $\alpha > 0$ that satisfy $(\frac{1}{d} + \alpha^p) + 2(\frac{1}{d} - \alpha^p) + \frac{1}{d}\left((\frac{1}{d} + \alpha)^p + (\frac{1}{d} + l\alpha)^p\right) > a^p - 3a^p + l\alpha^p$, by intermediate value theorem there must exist $\alpha > 0$ that satisfies inequality (5). Notice that regardless of the value of $l$, we could always find $\alpha > 0$ large enough so that the condition (a) is satisfied; and for condition (b), by having $\alpha = \sqrt{\dfrac{1}{d}}$, we could find $l > 0$ that satisfies the condition.

Now we have $2r + 2d$ voters in total, and they all vote for $c_1$. RVPM moves $c_1$ to $c_1$ so that $c_1$ loses at least $r + d$ voters and loses the election. Recall that $v_{-D}$ will always vote for $c_1$. We next find a solution $c_1$ that will lose $d$ votes in $\{v_{e_i}, v_{-e_i}\}_{i=1}^d$ and all voters in $\{v_i\}_{i=1}^r$.

Notice that the distance between $c_2$ and any voter in $\{v_{e_i}, v_{-e_i}\}_{i=1}^d$ is $(a^p + 1)^{\frac{1}{p}}$. We claim that for $c_1$ to lose $d$ votes in $\{v_{e_i}, v_{-e_i}\}_{i=1}^d$ (which is maximum), $c_1$ has to have format $\pm \frac{1}{d}, \pm \frac{1}{d}, \ldots, \pm \frac{1}{d}$. Given the budget constraint $\|c_1\|_p \leq \epsilon$, the solution of $c_1$ losing maximum number of voters can be achieved at the constraint boundary, since we can always move $c_1,d+1$ away from 0 until $\|c_1\|_p = \epsilon$ and this will only increase the distance between $c_1$ and any voter in $\{v_{e_i}, v_{-e_i}\}_{i=1}^d \cup \{v_i\}_{i=1}^r$. With $\|c_1\|_p = \epsilon$, we have $\|c_1 - v_{e_i}\|_p^p = \|c_1 - 1\|^p + d^{1-p} - \|c_1, i\|_p, \|c_1 - v_{-e_i}\|_p = \|c_1, l\|^p + d^{1-p} - \|c_1, i\|_p$. Recall that in destructive control $c_1$ will lose the voter in case of tie-breaking. Since for $c_1$ to lose $v_{e_i}$, due to the budget constraint, we must have $c_1,i \leq -\frac{1}{d}$; and for $c_1$ to lose $v_{-e_i}$, we must have $c_1,i \geq \frac{1}{d}$; $c_1$ can only lose maximum one voter between $v_{e_i}$ and $v_{-e_i}, i \in [d]$. Given the budget constraint, in order for $c_1$
to lose $d$ voters in $\{v_{c_i}, v_{c_i}\}^d_{i=1}$, $\tilde{c}_1$ has to have the above format.

Next we show that finding a $\tilde{c}_1$ to lose all $r$ voters in $\{v_i\}^r_{i=1}$ is equivalent to finding a solution to 3-SAT.

Given a 3-SAT solution $\{X_1, \ldots, X_{d-1}\}$, we let $c_{1,i} = \frac{1}{2}$ if $X_i$ is TRUE, $c_{1,i} = -\frac{1}{2}$ if $X_i$ is FALSE, $i \in [d-1]$; $c_{1,d} = -\frac{1}{2}$ and $c_{1,d+1} = 0$. It is easy to check $\tilde{c}_1$ satisfies the budget constraint. For any clause $E_i, i \in [r]$ that has $t$ literals to be TRUE ($1 \leq t \leq 3$), from inequality (5) we have $||\tilde{c}_1 - v_i||_p \geq ||c_2 - v_i||_p$, meaning $\tilde{c}_1$ loses voter $i$ and is a solution to RVPM.

Given a RVPM solution $\tilde{c}_1$, since $\tilde{c}_1$ has format $[\pm \frac{1}{2}, \pm \frac{1}{2}, \ldots, \pm \frac{1}{2}, \pm \frac{1}{2}, 0]$, we can construct a 3-SAT solution $\{X_1, \ldots, X_{d-1}\}$ where variables $X_i = TRUE$ if $c_{1,i} = \frac{1}{2}$, or $X_i = FALSE$ if $c_{1,i} = -\frac{1}{2}, i \in [d-1]$. Since $v_{i,d} = \alpha \alpha$, we assume $c_{1,i} = -\frac{1}{2}$. For any clause $E_i, i \in [r]$ that has $t$ literals to be TRUE ($0 \leq t \leq 3$), since we need $\tilde{c}_1$ to lose voter $v_i$, meaning $||\tilde{c}_1 - v_i||_p \geq ||c_2 - v_i||_p$, from inequality (5) we must have $t \geq 1$.

A.4 Proof of Theorem 3 ($l_\infty$ norm)

For RVPM $l_\infty$ norm, again, it is easy to see that it’s in NP. We now show hardness again by reduction from 3-SAT. The proof is in $\mathbb{R}^{d+1}$ space.

We set target candidate $c_1 = [0, 0, \ldots, 0, 0]$, rival candidate $c_2 = [0, 0, \ldots, 0, 2]$ and let $e = 1$. We are given a 3-SAT instance with $d - 1$ variables $\{X_1, X_2, \ldots, X_{d-1}\}$ and $r$ clauses $\{E_1, E_2, \ldots, E_r\}$. We create $r$ voters $\{v_i\}^r_{i=1}$ where each $v_i$ corresponds to clause $E_i, i \in [r]$. For each $v_i$, we set $v_{i,j} = 1$ if $X_j$ is included in clause $E_i$, or $v_{i,j} = 0$ if $X_j$ is included in clause $E_i, j \in [d-1]$; otherwise. We also create $r$ dummy voters at $[0, 0, \ldots, 0, 0]$; those dummy voters will always vote for $c_1$ since their distances to $c_2$ are always 2 while their distances to $c_1$ are within 1 whenever $c_1$ moves. We now have $2r$ voters, and they all vote for $c_1$. RVPM will change $c_1$ to $\tilde{c}_1$ so that $\tilde{c}_1$ loses at least $r$ voters.

Given a 3-SAT solution $\{X_1, \ldots, X_{d-1}\}$, we set $c_{1,i} = 1$ if $X_i$ is TRUE, and $c_{1,i} = -1$ if $X_i$ is FALSE, $i \in [d-1]$; $c_{1,d} = -1$ if $X_{d} = TRUE$ or $c_{1,d} = 1$ if $X_{d} = FALSE$. It is easy to check $||\tilde{c}_1 - v_i||_{\infty} = 2, \forall i \in [r]$. $\tilde{c}_1$ loses voters and loses the election.

Given a RVPM solution $\tilde{c}_1$, since $\tilde{c}_1$ loses the election, we must have $||\tilde{c}_1 - v_i||_{\infty} \geq ||c_2 - v_i||_{\infty} = 2, \forall i \in [r]$. Since $v_{i,j} \in \{0, \pm 1\}^d, \forall i \in [r]$ and $||c_1||_{\infty} \leq 1$, $\tilde{c}_1$ can only lose $r$ voters by having $||\tilde{c}_1 - v_i||_{\infty} = 2, \forall i \in [r]$. Since $v_{i,d} = 0$, we must have $\tilde{c}_{1,i} \in \{-1, 1\}, i \in [d - 1]$. By setting $X_i = TRUE$ if $\tilde{c}_{1,i} = 1, X_i = FALSE$ if $\tilde{c}_{1,i} = -1, i \in [d-1]$, we have a solution $\{X_1, \ldots, X_{d-1}\}$ to 3-SAT.

A.5 Proof of Theorem 4 ($l_p$ norm, $1 < p < \infty$)

Checking whether a given target candidate $\tilde{c}_1$ wins the election is clearly in P. We now show hardness by reduction from 3-SAT. The proof is in $\mathbb{R}^d$ space.

Let $c_i \in \mathbb{R}^d$ be the unit vector that has 1 in the $i$-th position, $d' = d - 1$. We first introduce the below lemma.

**Lemma 6.** The smallest ball that encloses $\{e_i\}^d_{i=1}$ is centered at $c \sum_{i=1}^{d'} e_i$, where $c = \frac{1}{1 + (d' - 1)^{1/(p-1)}}$, and the radius of the ball is $r_{d'} = ((d' - 1)c^p + (1 - c)^p)^{1/p}$.

**Proof.** By symmetry, the center of the smallest ball has the form $c \sum_{i=1}^{d'} e_i (0 \leq c \leq 1)$, and its distance to $e_i, i \in [d']$ is $||e_i - c \sum_{i=1}^{d'} e_i||_p = ((d' - 1)c^p + (1 - c)^p)^{1/p}$.

Since the second order derivative of the distance w.r.t. $c$ is always greater than 0, by having the first order derivative equals zero, we have $c = \frac{1}{1 + (d' - 1)^{1/(p-1)}}$.

Notice that we could always change any $e_i$ to $-e_i, i \in [d']$ in Lemma 6 the conclusion still holds.

Next we construct the voter set and candidate set for RVPM problem. Let target candidate $c_T = [0, 0, \ldots, 0, 0]$ and $e = c \cdot (d')^b$ ($c$ is defined in Lemma 6). Given a 3-SAT instance with $d - 2$ variables $\{X_1, \ldots, X_{d-2}\}$ and $r$ clauses $\{E_1, E_2, \ldots, E_r\}$, we create $r$ voters $\{v_i\}^r_{i=1}$ and $r$ candidates $\{c_i\}^r_{i=1}$ where each $v_i$ and $c_i$ corresponds to clause $E_i, i \in [r]$. Given fixed parameters $l > 0, \alpha > 0$ (which we will specify later), if $X_j$ is included in clause $E_i$, we set $v_{i,j} = c_{i,j} = \alpha, \forall i \in [d - 2]$; $v_{i,d} = c_{i,d} = \alpha; c_{i,d} = r_{d'}$ (r_{d'} is defined in Lemma 6); 0 otherwise. We also create $r$ dummy voters at $[0, 0, \ldots, 0, 0]$; those dummy voters will always vote for $c_1$ since their distances to $c_2$ are always 2 while their distances to $c_1$ are within 1 whenever $c_1$ moves. We now have $2r$ voters, and they all vote for $c_1$. RVPM will change $c_1$ to $\tilde{c}_1$ so that $\tilde{c}_1$ wins at least $r$ voters and wins the election.

Recall that when we create the voters and candidates w.r.t. 3-SAT clauses, we have two fixed parameters $l > 0, \alpha > 0$. We chose those two parameters to satisfy the below inequality:

$$|\alpha - c|^p + 2(\alpha + c)^p + (d' - 4)c^p + |\alpha - c|^p < r_{d'}^p < 3(\alpha + c)^p + (d' - 4)c^p + |\alpha - c|^p.$$  

We claim that parameters $l > 0, \alpha > 0$ that satisfy inequality (6) must exist. Since $d'$ and $p$ are given, $r_{d'}$ is a fixed value in $\mathbb{R}$, and the right and left side of the inequality are continuous w.r.t. $\alpha$ and $l$, with gap equals to $(\alpha + c)^p - |\alpha - c|^p > 0$. This means if we could find (a) $l > 0, \alpha > 0$ that satisfy $|\alpha_1 - c|^p + 2(\alpha_1 + c)^p + (d' - 4)c^p + |\alpha_1 - c|^p > r_{d'}^p$, and (b) $\alpha_2 > 0$ that satisfy $|\alpha_2 - c|^p + 2(\alpha_2 + c)^p + (d' - 4)c^p + |\alpha_2 - c|^p < r_{d'}^p$, by intermediate value theorem there must exist $\alpha > 0$ that satisfies inequality (6). Notice that regardless of the value of $l$, we could always find $\alpha_1 > 0$ large enough so that condition (a) is satisfied. Next we find $l > 0$ and $\alpha_2 > 0$ that satisfy condition (b). Plug in $r_{d'}^p = (d' - 1)c^p + (1 - c)^p$, we have $|\alpha_2 - c|^p < 3c^p + (1 - c)^p - |\alpha_2 - c|^p - 2(\alpha_2 + c)^p$. Since $0 < c < \frac{1}{2}$, there exists $\alpha_2$ such that $3c^p + (1 - c)^p - |\alpha_2 - c|^p - 2(\alpha_2 + c)^p$.  


\[ |\alpha_2 - c|^p - 2|\alpha_2 + c|^p > 0, \text{ we can then find } l > 0 \text{ such that} \]
\[ \text{the inequality holds (e.g., } l = c/|\alpha_2|). \]

With parameters \( l > 0 \) and \( \alpha > 0 \) chosen and fixed, next we show that finding a \( \tilde{c} \) to win the election (i.e., win at least \( r + d' \) voters) is equivalent to finding a solution to 3-SAT.

Given a 3-SAT solution \( \{X_1, \ldots, X_{d-2}\} \), we let \( \tilde{c}_{T,i} = c \) if \( X_i \) is TRUE, \( \tilde{c}_{T,i} = -c \) if \( X_i \) is FALSE, \( i \in [d-2]; \)
\[ \tilde{c}_{T,d'} = c; \tilde{c}_{T,d} = 0. \]

It is easy to check \( \tilde{c}_T \) satisfies the budget constraint and win \( d' \) voters in \( \{v_{e_1}, \ldots, v_d\} \) by having a distance of \( r_{d'} \). For any clause \( E_i, i \in [r] \) that has \( t \) literals to be TRUE \( (1 \leq t \leq 3) \), by inequality (6) we have
\[ ||v_i - \tilde{c}_T||_p \]
\[ = t|\alpha - c|^p + (3 - t)(\alpha + c)p + (d' - 4)c^p + |\alpha - c|^p < r_{d'}. \]

\( \tilde{c}_T \) wins \( r + d' \) voters and wins the election.

Given a RVPM solution \( \tilde{c}_T \), since \( \tilde{c}_T \) needs to win at least \( r + d' \) voters, this means \( \tilde{c}_T \) has to have a distance of no more than \( r_{d'} \) to at least \( d' \) voters in \( \{v_{e_1}, \ldots, v_d\} \). By Lemma 6 and budget constraint requirement, \( \tilde{c}_T \) has to have format \( \tilde{c}_{T,i} \in \{-c, c\}, i \in [d-1], \tilde{c}_{T,d'} = 0, \) and \( \tilde{c}_T \) wins only \( d' \) voters in \( \{v_{e_1}, \ldots, v_{d'}\} \). Since \( v_{d'} = \alpha \alpha \), \( i \in [r] \), we assume \( \tilde{c}_{T,d'} = c \). We can then construct a 3-SAT solution \( \{X_1, \ldots, X_{d-2}\} \) where variables \( X_i \) is TRUE if \( \tilde{c}_{T,i} = c, X_i = \text{ FALSE if } \tilde{c}_{T,i} = -c, i \in [d-2]. \)

For any clause \( E_i, i \in [r] \) that has \( t \) literals to be TRUE \( (0 \leq t \leq 3) \), the distance between \( v_i \) and \( \tilde{c}_T \) is
\[ ||v_i - \tilde{c}_T||_p \]
\[ = t|\alpha - c|^p + (3 - t)|\alpha + c|^p + (d' - 4)c^p + |\alpha - c|^p. \]

Since \( \tilde{c}_T \) needs to win all \( r \) voters in \( \{v_i\} \), we must have \( ||v_i - \tilde{c}_T||_p \leq r_{d'} \) by inequality (6), this implies \( t \geq 1 \).

This proof is based on the covering by two balls problem in Megiddo [1990]. While the original problem is under \( l_2 \) norm, we generalize it to \( l_p \) with \( p \in (1, \infty) \).

A.6 Proof of Theorem 4 (\( l_{\infty} \) norm)

First, it is easy to see that the problem is in NP. We now show hardness by reduction from 3-SAT. The proof is in \( \mathbb{R}^d \) space.

We set target candidate \( c_1 = [0, 0, \ldots, 0, 0] \) and \( \epsilon = 1/2 \). We are given a 3-SAT instance with \( d - 1 \) variables \( \{X_1, X_2, \ldots, X_{d-1}\} \) and \( r \) clauses \( \{E_1, E_2, \ldots, E_r\} \). For each clause, we create \( 7 \) sets of voters and candidates that each corresponds to a solution to the clause to be TRUE. For example, for clause \( (X_i \vee \overline{X_j} \vee X_k) \) to be TRUE, we can have
\[ X_i = 1, X_j = 1, X_k = 1 \]
\[ X_i = 1, X_j = 1, X_k = -1 \]
\[ X_i = 1, X_j = -1, X_k = 1 \]
\[ X_i = 1, X_j = -1, X_k = -1 \]
\[ X_i = -1, X_j = 1, X_k = 1 \]
\[ X_i = -1, X_j = 1, X_k = -1 \]
\[ X_i = -1, X_j = -1, X_k = 1 \]
\[ X_i = -1, X_j = -1, X_k = -1 \]

where 1 represents TRUE and -1 represents FALSE. For each such solution, we create a voter and a candidate where
\[ v_{i,j} = c_{i,j} = 1 \text{ if } X_i = 1, \text{ or } v_{i,j} = c_{i,j} = -1 \text{ if } X_i = -1, i \in [d-1]; c_{d} = 1/2; 0 \text{ otherwise.} \]

The key idea is to move \( c_{i,j} \) as close to \( c_{i,j} \) as possible for each issue \( j \in [d], \) leaving minimum gaps. Given the voter set \( V, c_1, c_2 \) and \( \epsilon \), the algorithm is as below:
\[ c_{i,j} = \text{sign}(c_{i,j} - c_{i,j}) \cdot \min(|c_{i,j} - c_{i,j}|, \epsilon) + c_{i,j}, j \in [d]. \]

If \( c_1 \) wins the election, return \( c_1 \); else return NO.

Next we give a formal proof of the algorithm. Since under \( l_{\infty} \) norm only the relative distance between \( c_{i,j} \) and \( c_{i,j} \) matters. Without loss of generality, we assume \( c_1 = [0, 0, \ldots, 0], c_2 = [a_1, a_2, \ldots, a_d]. \) Since the issues are independent from one another, we permute the issues so that \( 0 < |a_1| \leq |a_2| \leq \ldots \leq |a_d| \) (for issues that \( c_1 \) and \( c_2 \) agree on, i.e., \( a_j = 0 \), we can simply omit them). Given budget \( \epsilon \), we assume \( |a_i| \leq \epsilon < |a_{i+1}|. \) We claim that \( c_1 = [a_1, a_2, \ldots, a_d] \) wins maximum number of voters possible within budget.

Let \( c_i' = [a_1, a_2, \ldots, a_d] \) be a point within \( \epsilon \) from \( c_1 \) that wins maximum number of voters possible. For any voter \( v_r \in V \) that votes for \( c_1 \), assume \( \max(v_r, c_2) = |v_r, a_k| \). To start with, we set \( c_1 = c_1' \), then we have \( \max(v_r, c_1) = \max(|v_r, a_1 - a_1|, \ldots, |v_r, a_d - a_d|) \).

Notice that since \( |v_r, k - a_k| \geq |v_r, j - a_j|, j \in [d], \) and \( |a_j| \leq \epsilon, j \in [i], \) we can set \( c_{i,j} = a_j \) for \( j \in [i] \) and this has no impact on \( c_1 \) winning voter \( v_r. \) Now \( c_1 = [a_1, a_2, \ldots, a_{i+1}, \ldots, a_d]. \) Next we only need to discuss the values of \( c_{i,j} \) for \( j \in [i + 1, \ldots, d]. \) We show that \( |v_r, k - a_k| \geq |v_r, j - a_j| \) for all
Due to the budget constraint to $\tilde{c}_i$, we must have $-\epsilon \leq a_j \leq \epsilon, \forall j \in [d]$. For $\forall j \in \{i+1, \ldots, d\}$, we have:

- if $a_j > 0$:
  - if $v_{r,j} \geq \epsilon > a_j$, then $|v_{r,k} - a_k| \geq |v_{r,j} - a_j| = v_{r,j} - a_j \geq \epsilon = |v_{r,j} - \epsilon|$
  - if $v_{r,j} > a_j$, then since $a_j > \epsilon$, we have $v_{r,k} - a_k \geq |v_{r,j} - a_j| = a_j - v_{r,j} > \epsilon - v_{r,j} = |v_{r,j} - \epsilon|$

- if $a_j < 0$:
  - if $a_j \geq v_{r,j} \geq a_j$, then $|v_{r,k} - a_k| \geq |v_{r,j} - a_j| = a_j - v_{r,j} \geq \epsilon - v_{r,j} = |v_{r,j} + \epsilon|$
  - if $a_j \geq v_{r,j}$, then $|v_{r,k} - a_k| \geq |v_{r,j} - a_j| = v_{r,j} - a_j \geq \epsilon - v_{r,j} = |v_{r,j} + \epsilon|$.

This means $\tilde{c}_i = [a_1, \ldots, a_i, \text{sign}(a_{i+1})\epsilon, \ldots, \text{sign}(a_d)\epsilon)$ wins voter $v_j$. Since $\tilde{c}_i$ wins maximum number of voters possible, $\tilde{c}_i$ also wins maximum number of voters possible. Next we only need to check whether $\tilde{c}_1$ wins the election (i.e., wins at least half of the voters).

The computation of $\tilde{c}_1$ takes $O(d)$ time; checking whether $\tilde{c}_1$ wins the election takes $O(md)$ time. The algorithm takes $O(mn\log n)$ time in total.

### A.8 Proof of Theorem 7

We solve the problem by finding the set which contains a representative solution point for all scenarios of $(t_1, \ldots, t_m)$ (defined as in Theorem 6), which covers all the scenarios of $c_1$ winning. Each scenario represents finding $\tilde{c}_1$ within budget constraint that gets a score of at least $f(t_j)$ from voter $v_j$, $j \in [m]$, and is to solve the below feasibility problem:

$$||\tilde{c}_1 - c_1||_\infty \leq \epsilon$$

where $d_j$ is defined in the constructive control variant of BVPM. There are in total $m \cdot |P_{\text{min}}| + 1$ hyperspheres involved for all the scenarios. We hope to find the representative points of intersections for those $m \cdot |P_{\text{min}}| + 1$ hyperspheres which then cover all the scenarios of $c_1$ winning. As discussed in Crama et al. [1995], the problem goes down to box intersection problem for constant $d$. For a problem that involves $n$ boxes, it can be solved in time $O(n \log n)$ for $d \leq 2$ and $O(n^{d-1})$ for $d \geq 3$ [Imai and Asano, 1983; Lee, 1983]. The resulting algorithm is exponentially dominant in $d$.

### A.9 Proof of Lemma 3

We denote $S_j$ as the boundary of $B_j$, and $S_j$ as the boundary of both closed ball $B_j$ and open ball $B_j$, $j \in [k]$.

As defined in Theorem 5 of Crama et al. [1995], given $\{B_1, B_1, \ldots, B_k\}$ in $\mathbb{R}^d$, let $P$ be a set of points in $\mathbb{R}^d$ where for each $F \subseteq \{B_1, B_1, \ldots, B_k\}$ with $|F| \leq d$,

(i) if $\bigcap_{j \in F} S_j$ is connected, then $P$ contains a point of $\bigcap_{j \in F} S_j$.

(ii) if $\bigcap_{j \in F} S_j$ contains at most two points, then $P$ contains $\bigcap_{j \in F} S_j$.

Assume the solution set $P' \neq \emptyset$. We chose $x \in P'$ as a representative solution. Let $H = \{j \in \{1, \ldots, k\} \mid x \in B_j\}$, $F = \{j \in \{1, \ldots, k\} \mid x \in S_j\}$. Let $x$ be chosen so that first $|H|$ is as small as possible, then $|F|$ is as large as possible. According to Lemma 5 in Crama et al. [1995], the intersections of spheres are either connected or contain at most two points. There are four cases to discuss:

Case 1: $\bigcap_{j \in F} S_j$ is connected, and $|F| < d$. By construction of $P$, there exists a point $u \in P \cap \bigcap_{j \in F} S_j$. We next claim that $u \in B_c$, and for all $j \in \{1, \ldots, k\}$, $u \notin B_j$. Since $x$ is a solution, we have $\{i_1, \ldots, i_r\} \cap H = \emptyset$, this also implies $u \notin B_j, j \in \{i_1, \ldots, i_r\}$. Since $\bigcap_{j \in F} S_j$ is connected and both $x, u \in \bigcap_{j \in F} S_j$, there is a path from $x$ to $u$ on $\bigcap_{j \in F} S_j$. If the claim is not valid, then moving from $x$ to $u$ we must encounter a first boundary of the ball $S_j$ with index $j \in \{1, \ldots, k\}$. Let $b$ be the point that the path from $x$ to $u$ intersects with $S_j$. If $j \in H$, this contradicts the minimality of $H$, since $v$ is in fewer balls than $x$; if $j \notin H$, this contradicts the maximality of $F$, since $v$ is on more spheres than $x$.

Case 2: $\bigcap_{j \in F} S_j$ is connected, and $|F| \geq d$. According to Lemma 5 in Crama et al. [1995], there exists $F', |F'| < d$, such that $\bigcap_{j \in F} S_j = \bigcap_{j \in F'} S_j$. This goes to case 1.

Case 3: $\bigcap_{j \in F} S_j$ contains at most two points, $|F| \leq d$. By construction, $P$ contains $\bigcap_{j \in F} S_j$, and $x \in \bigcap_{j \in F} S_j \subseteq P$.

Case 4: $\bigcap_{j \in F} S_j$ contains at most two points, $|F| > d$. We take $F' \subseteq F$, $|F'| = d$. Since the intersections of $d$ non-coinciding spheres have at most two points, by construction $P$ contains $\bigcap_{j \in F} S_j$, and $x \in \bigcap_{j \in F} S_j \subseteq \bigcap_{j \in F'} S_j \subseteq P$.

The proof is modified based on Lemma 3 in Crama and Ibaraki [1997], where it studies the avoid ball problem in $\mathbb{R}^d$ for constant $d$ and the feasibility area is bounded by a $d$-dimensional hypersphere.

### A.10 Proof of Lemma 4

We claim that if $\tilde{y} \in \mathbb{R}^d$ satisfy constraint (2a) and $\bigcup_{j=1}^d S_j(\tilde{y}_j) = [k]$, then $\tilde{y}$ is a solution to the feasibility problem. If the claim is not true, assume the $l$-th constraint of (2b) is not satisfied for some $l \in [k]$, meaning $||\tilde{y} - a_l||_\infty < b_l$, this implies $|\tilde{y}_j - a_{l,j}| < b_l, \forall j \in [d]$ and $l \neq S_j(\tilde{y}_j), \forall j \in [d]$, which contradicts $\bigcup_{j=1}^d S_j(\tilde{y}_j) = [k]$.

Given the above result, we next show for a solution $\tilde{y}$, its $j$-th dimension coordinate $\tilde{y}_j \in [-\epsilon + y_j, \epsilon + y_j]$ can be represented by one of the points in $P_j$. Notice that for any $i$-th constraint in (2b), $i \in [k]$, its $j$-th dimension $|\tilde{y}_j - a_{i,j}| < b_i$ depicts an open interval on $\mathbb{R}^d$: $[-b_i + a_{i,j}, b_i + a_{i,j}]$. The $j$-th dimension of $k$ constraints correspond to $k$ open intervals. Since the intervals are open, $y_j$ and its nearest open interval endpoint $\tilde{y}_j \in [-\epsilon + y_j, \epsilon + y_j]$ satisfies $S_j(\tilde{y}_j) = S_j(\tilde{y}_j')$. If there is no open interval endpoint within $[-\epsilon + y_j, \epsilon + y_j]$, then any $\tilde{y}_j$ that is within $[-\epsilon + y_j, \epsilon + y_j]$ satisfies $S_j(\tilde{y}_j) = S_j(\tilde{y}_j')$ (e.g., $\tilde{y}_j' = y_j$). This means $\bigcup_{j=1}^d S_j(\tilde{y}_j) \cup \bigcup_{j=1}^d S_j(\tilde{y}_j')$ contains all such $\tilde{y}_j', \forall j \in [d]$, if the problem is feasible then
For each voter \(v\), the score each candidate receives from \(c\) candidates to \(v\) is as below:

\[
\begin{array}{cccccccc}
\text{f}(1) & \ldots & \text{f}(r_j-1) & \text{f}(r_j) & \text{f}(r_j+1) & \ldots & \text{f}(n)
\end{array}
\]

If we move \(c_1\) from ranking position \(r_j\) to \(r_j'\), \(r_j' \geq r_j\), since \(f\) is a non-increasing function, the score \(c_1\) receives from \(v\) will decrease from \(f(r_j)\) to \(f(r_j')\); the scores candidates \(c_n, c_{n-1}, \ldots, c_{n-r_j+2}\) and \(c_{n-r_j'+1}, c_{n-r_j'}, \ldots, c_2\) receive from \(v\) will not change; for candidates \(c_{n-r_j+1}, c_{n-r_j}, \ldots, c_{n-r_j'+2}\), since their rankings will increase by 1, the scores they receive from \(v\) will not decrease. This means if we move \(c_1\) from ranking position \((r_1, r_2, \ldots, r_m)\) to \((r_1', r_2', \ldots, r_m')\), \(r_j' \geq r_j, \forall j \in [m]\), the total score \(c_1\) receives will not increase, while the total scores other candidates receive will not decrease. Since \(c_1\) loses the election with ranking position \((r_1', r_2', \ldots, r_m')\), \(c_1\) will also lose the election with ranking position \((r_1, r_2, \ldots, r_m)\). \(\square\)

**Lemma 8.** For destructive control, by enumerating all scenarios of \(c_1\) getting a ranking position lower than \(t_j\) w.r.t. voter \(v_j\), for \(t_j \in \{s_1, \ldots, s_r\}, \forall j \in [m]\), we cover all the possible scenarios of \(c_1\) losing.

**Proof.** Without loss of generality, we assume candidates \(c_{n-2}, c_{n-1}, c_2\) are ranked from closest to furthest w.r.t. voter \(v_j\). We next insert \(c_1\) into this sequence and assume \(c_1\) receives a score of \(f(s_1)\) from \(v_j\), where \(i \in \{1, \ldots, r\}\). After the insertion, the score each candidate receives from \(v_j\) is as below:

\[
\begin{array}{cccc}
\text{f}(s_{i-1}) & \text{f}(s_i) & \text{f}(s_{i+1}) & \text{f}(s_{i+2})
\end{array}
\]

Notice that whichever ranking position \(c_1\) takes among \(s_i, s_i+1, \ldots, s_i+1\), the final score each candidate receives from \(v_j\) are the same, meaning these ranking positions are equivalent. Since the solution set of getting a ranking position lower than \(s_i\) covers the solution set of getting a ranking position lower than \(s_i+1, s_i+2, \ldots, s_i+1\), we only need to consider \(s_i\). \(\square\)

**C Algorithm in Lemma 5**

**Algorithm 1** Feasibility problem with constant number of constraints

**Input:** Feasibility problem parameters.

**Output:** A solution to the problem; or NO if not feasible.

1: Calculate \(S_1\) and \(P_1\)
2: \(S \leftarrow S_1, P \leftarrow P_1\)
3: for \(j \in [2 : d]\) do
4: Calculate \(S_j\) and \(P_j\)
5: Find all pairs of sets \(S(p) \in S\) and \(S_j(p_j) \in S_j\), where \(S_j(p_j) \not\subseteq S(p)\).
6: Add \(S(p) \cup S_j(p_j)\) to \(S\) and \([p, p_j]\) to \(P\)
7: if \([k] \in S\) then
8: Expand its corresponding point \(p\) to \(d\)-dimension
9: return \(p\)
10: end if
11: Remove subsets in \(S\) and corresponding points in \(P\)
12: Update \(P\) so that all points have dimension \(j\)
13: end for
14: return NO

For simplicity, we first define set \(S_j = \{S_j(p_j) \mid p_j \in P_1\}\), where \(P_1\) and \(S_j(p_j)\) are defined in Lemma 4. Assume a point \(p\) is \(j\)-dimensional, we let \(S(p) = \cup_{j=1}^d S_j(p_i)\). According to Lemma 4, finding a solution to the feasibility problem is equivalent to finding a point \(p \in \mathbb{R}^d\) that satisfy constraint (2a) and \(S(p) = [k]\).

In Line 1 and 2, we initialize the sets \(S\) and \(P\) with \(S_1\) and \(P_1\), and later use them to store the results for the \([1 : j]\) dimension. In the main loop, for each dimension \(j \in [2 : d]\), we first calculate sets \(S_j\) and \(P_j\), and then find all pairs of sets
\( S(p) \in S \) and \( S_j(p_j) \in S_j \) that satisfy \( S_j(p_j) \not\subseteq S(p) \). This means \( S(p) \subset S([p, p_j]) \). We add all such sets \( S([p, p_j]) \) to \( S \) and points \([p, p_j] \) to \( P \). The terminate condition is if \([k] \in S \), then for the \( j \)-dimensional point \( p \in P \) that has \( S(p) = [k] \), \( S([p, y_{j+1}, \ldots, y_d]) = [k] \) and \([p, y_{j+1}, \ldots, y_d] \) satisfies constraint (2a). \([p, y_{j+1}, \ldots, y_d] \) is a solution to the feasibility problem.

In Line 11 we remove all subsets in \( S \) (as well as the corresponding points in \( P \)) and keep the sets in \( S \) pairwise incomparable. This improves our algorithm efficiency and has no impact on us finding a solution: if a solution \( \tilde{y} \) exists with \( \cup_{i=1}^{d} S_i(y_i) = [k] \), and we have \( y'_j \in [-\epsilon + y_j, \epsilon + y_j] \) with \( S_j(y_j) \subseteq S_j(y'_j) \), we then have \( \cup_{i=1}^{j-1} S_i(y_i) \cup S_j(y'_j) \cup_{i=j+1}^{d} S_i(y_i) = [k] \), meaning \([y_1, \ldots, y_{j-1}, y'_j, y_j+1, \ldots, y_d] \) is also a solution.

In Line 12, we append \( y_j \) to all the \((j-1)\)-dimensional points in \( P \) so that all points are \( j \)-dimensional.

Since \( k \) is constant, the sizes of \( S \) and \( S_j \) as well as their elements are all constants. The algorithm is linear and of complexity \( O(d) \). We can also determine in linear time if none of the representative points satisfy the feasibility condition, and return NO in that case.