Apportionment with Thresholds: Strategic Campaigns Are Easy in the Top-Choice But Hard in the Second-Chance Mode

Christian Laußmann, Jörg Rothe, Tessa Seeger

Heinrich-Heine-Universität Düsseldorf, MNF, Institut für Informatik
{christian.laussmann, rothe, tessa.seeger}@hhu.de

Abstract

In apportionment elections, a fixed number of seats in a parliament are distributed to parties according to their vote counts. Common procedures are divisor sequence methods like D'Hondt or Sainte-Laguë. In many countries, an electoral threshold is used to prevent very small parties from entering the parliament. Parties with fewer than a given number of votes are simply removed. We (experimentally) show that by exploiting this threshold, the effectiveness of strategic campaigns (where an external agent seeks to change the outcome by bribing voters) can be increased significantly, and prove that it is computationally easy to determine the required actions. To resolve this, we propose an alternative second-chance mode where voters of parties below the threshold receive a second chance to vote for another party. We establish complexity results showing that this makes elections more resistant to strategic campaigns.

1 Introduction

In parliamentary elections, votes are cast for parties which in turn compete for a fixed number of seats in parliament. An apportionment method is then used to apportion the seats to the parties according to their vote counts. Usually, such methods aim at apportioning the seats in a way that makes the parliament form a small but somehow proportional representation of the voters. Such a representative parliament can then efficiently discuss topics and decide laws in the name of the voters. In many countries, the basic procedure is extended by a so-called *legal electoral threshold* (simply threshold, for short)—a minimum number of votes a party must receive to participate in the apportionment process at all. For instance, in Germany, Poland, and Scotland a party must receive at least 5% of the total vote count to participate in the apportionment process. Electoral thresholds are important for the government to quickly form and allow for effective decision-making by minimizing the effects of fragmentation of the parliament, i.e., by reducing the number of parties in it (see Pellicer and Wegner (2014) for a study of how mechanical and psychological effects reduce fragmentation). Undoubtedly, with fewer parties in the parliament compromises can be made more efficiently.

However, a disadvantage of the threshold is that voters supporting a party that did not make it above the threshold are not represented in the parliament at all because their votes are simply ignored. For example, more than 19% of the votes in the French election of the European Parliament in 2019 were lost due to a threshold of 5%, i.e., *"of five votes, just four become effective, and one is discarded as ineffective"* (Oelbermann and Pukelsheim 2020, p. 30).

Apart from these benefits and drawbacks of thresholds, we want to find out to what extent they can be exploited in *strategic campaigns*. In such scenarios, an external agent intends to change the election outcome in her favor by bribing voters within a certain budget to change their vote. That is, an external agent seeks to change a minimum number of votes in order to either ensure a party she supports receives at least ℓ seats in the parliament (constructive case), or to limit the influence of a party she despises by ensuring it receives no more than ℓ seats (destructive case). Today's possibilities to process enormous amounts of data from social networks, search engines, etc., make it possible to predict the voting behavior of individuals and to target them with individualized (political) advertising. Because of this, strategic campaigns attract increasing attention in political elections. Given that these attempts are already being used in the real world, it is critical to understand the threat they pose. To assess these risks, it is essential to know how effective such campaigns can be and how easy it is to find optimal campaigns. Additionally, if there is a high risk, it is desirable to improve apportionment procedures to make them more resistant to such campaigns.

Related Work. The research line on *bribery*, a.k.a. *strategic campaigns*, was initiated by Faliszewski, Hemaspaandra, and Hemaspaandra (2009). For more background, we refer to the book chapters by Faliszewski and Rothe (2016) and Baumeister and Rothe (2015).

Bribery is linked to *electoral voter control* where voters are added to or deleted from voting registers by the election chair (see Bartholdi III, Tovey, and Trick (1992); Hemaspaandra, Hemaspaandra, and Rothe (2007)). Bribery and control have been studied for a wide range of voting rules, as surveyed by Faliszewski and Rothe (2016). While they have mainly been investigated for single-winner and multiwinner voting rules, Bredereck et al. (2020) only recently initiated the study of bribery in apportionment elections. They show that an optimal strategic campaign for apportionment elections *without* a threshold can be computed in polynomial time. Their study is most closely related to our work. For the more general study of apportionment methods in mathematical and political context, we refer to the works by Pukelsheim (2017) and Balinski and Young (1975, 1982).

Our Contribution. Our first contribution is incremental to the work of Bredereck et al. (2020): We adapt their algorithm to run for apportionment elections *with* thresholds, and to use binary search techniques for significantly accelerating the computation. Both improvements are important for the practical usability of the algorithms since most real-world (parliamentary) apportionment elections include a threshold, and the vote count is often in the order of 10 million, so algorithmic efficiency is highly important. By testing the improved algorithm on real-world elections we observe that the campaigns can exploit the electoral threshold and significantly benefit from it. Further, we introduce the destructive variant of strategic campaigns.

Our second and completely novel contribution is a simple extension of the usual apportionment procedure with electoral threshold: Voters who supported a party below the threshold can reuse their vote for one of the remaining parties above the threshold. These voters thus get a second chance. We provide complexity results showing that this modification makes optimizing the corresponding strategic campaigns intractable, thus protecting the election.

2 Preliminaries

We begin by introducing the following two notations. Throughout this paper, we denote the set $\mathcal{P} \setminus \{X\}$ by $\mathcal{P}_{-X}$, and we write $[x]$ as a shorthand for $\{1, \ldots, x\}$.

We now turn to the apportionment setting. There already exist simpler definitions of apportionment instances in the literature, but to treat the electoral threshold and the extension which we will propose in Section 5 conveniently, we propose the following definition that is close to the classical ones from single-winner and multiwinner voting and works in a two-stage process. An apportionment instance $I = (\mathcal{P}, \mathcal{V}, \tau, \kappa)$ consists of the set of m *parties* $\mathcal{P}$, a list of n *votes* $\mathcal{V}$ over the parties in $\mathcal{P}$, a *threshold* $\tau \in \mathbb{N} = \{0, 1, 2, \ldots\}$, and the *seat count* $\kappa \in \mathbb{N}$. Each vote in $\mathcal{V}$ is a strict ranking of the parties from most to least preferred, and we write $A \succ_v B$ if voter v prefers party A to B (where we omit the subscript v when it is clear from the context). We sometimes refer to the most preferred party of a voter v as v's *top choice*. We make the very natural assumption that we have more votes than we have both parties and seats. Note that in reality an electoral threshold is usually given as a relative threshold in percent (e.g., a 5% threshold). However, we can easily convert such a relative threshold into an absolute threshold, as required by our definition.[1]

An apportionment instance will be processed in two steps: First, we compute a *support allocation* σ, then we compute a *seat allocation* α. The support allocation $\sigma : \mathcal{P} \to \mathbb{N}$ describes how many voters support each party. Depending on this support, the parties later receive a corresponding number of seats in the parliament. In classical apportionment settings (which we consider in Sections 3 and 4), the support for each party is simply the number of top choices the party receives if the party receives at least τ top choices, otherwise, the support is 0. That is, votes for parties that receive less than τ top choices are ignored, and the voters have no opportunity to change their vote. We refer to this as the *top-choice mode*.[2] An alternative mode will be proposed in Section 5.

Example 1 (Support Allocation). *Consider* $\tau = 10$, $\mathcal{P} = \{A, B, C, D\}$, *and*

$$\begin{aligned}\mathcal{V} = (\quad & 8\times \quad A \succ B \succ C \succ D, && 12 \times B \succ A \succ C \succ D,\\ & 5\times \quad B \succ D \succ C \succ A, && 25 \times C \succ A \succ D \succ B,\\ & 10\times \quad D \succ B \succ A \succ C).\end{aligned}$$

In the top-choice mode, we obtain $\sigma(A) = 0$, $\sigma(B) = 17$, $\sigma(C) = 25$, *and* $\sigma(D) = 10$.

Given a support allocation, we can now determine the seat allocation by employing an *apportionment method*. As input, such a method takes the support allocation σ and the seat count κ, and computes the seat allocation $\alpha : \mathcal{P} \to \{0, \ldots, \kappa\}$ satisfying $\sum_{A \in \mathcal{P}} \alpha(A) = \kappa$. Note that the threshold does not matter for apportionment methods because it was already applied in the computation of the support allocation. In this study, we focus on the class of *divisor sequence apportionment methods* including, for example, the D'Hondt method (also known as the Jefferson's method), and the Sainte-Laguë method (also known as the Webster method). A divisor sequence method is defined by a sequence $d = (d_1, d_2, \ldots, d_\kappa) \in \mathbb{R}^\kappa$ with $d_i < d_j$ for all $i, j \in \{1, \ldots, \kappa\}$ with $i < j$, and $d_1 \geq 1$. For each party $P \in \mathcal{P}$, we compute the *fraction list* $\left[\frac{\sigma(P)}{d_1}, \frac{\sigma(P)}{d_2}, \ldots, \frac{\sigma(P)}{d_\kappa}\right]$. Then we go through the fraction lists of all parties to find the highest κ values (where ties are broken by some tie-breaking mechanism). Each party receives one seat for each of its list values that is among the κ highest values. D'Hondt is defined by the sequence $(1, 2, 3, \ldots)$ and Sainte Laguë is defined by $(1, 3, 5, \ldots)$.

Example 2 (D'Hondt). *Suppose we allocate* $\kappa = 6$ *seats to the parties, party* 1 *has support* 1104*, party* 2 *has* 363*, party* 3 *has* 355*, and party* 4 *has* 178*. Then, the resulting D'Hondt fraction lists are:*

$$\begin{array}{lcllllll} \textit{party}\ 1 & : & [\mathbf{1104}, & \mathbf{552}, & \mathbf{368}, & \mathbf{276}, & 220.8, & 184\ ],\\ \textit{party}\ 2 & : & [\ \mathbf{363}, & 181.5, & 121, & 90.8, & 72.6, & 60.5],\\ \textit{party}\ 3 & : & [\ \mathbf{355}, & 177.5, & 118.3, & 88.8, & 71, & 59.2],\\ \textit{party}\ 4 & : & [\ 178, & 89, & 59.3, & 44.5, & 35.6, & 29.7]. \end{array}$$

The $\kappa = 6$ *highest values are highlighted in boldface. Party* 1 *thus receives four seats, parties* 2 *and* 3 *receive one seat each, and party* 4 *receives no seats at all.*

Note that it is possible that a party does not receive any seats in parliament when an allocation procedure such

[1] Note that in strategic campaigns, as we define them, the total number of voters never changes but only which party they vote for. Thus the threshold is also constant.

[2] Note that in the top-choice mode it would be sufficient to know the top choice of each voter. However, we need the full preference later in the second-chance mode. So for convenience, we assume complete rankings for both modes.

as D'Hondt or Sainte-Laguë is applied, even if it receives enough votes to exceed the threshold, i.e., we only have the implication stating that if a party does not get enough votes to pass the threshold, it will not get any seats. However, the reverse implication does not apply. This can also be seen in the previous example, where party 4 has support greater than zero, i.e., exceeds the threshold, but it does not receive any seats when D'Hondt is applied.

Now we define strategic campaigns, modeled as a bribery scenario. We are given an apportionment instance, a budget K, and a number ℓ indicating the minimum number of seats we want to achieve for a distinguished party P^*. By bribing at most K voters to change their vote in our favor (i.e., we can alter their votes as we like), we seek to ensure that party P^* receives at least ℓ seats. To study whether finding successful campaigns (and checking whether there exist any at all) is tractable, we define the following decision problem (see Bredereck et al. (2020); Faliszewski, Hemaspaandra, and Hemaspaandra (2009)).

$\mathcal{R}$-THRESHOLD-APPORTIONMENT-BRIBERY	
Given:	An apportionment instance $(\mathcal{P}, \mathcal{V}, \tau, \kappa)$, a distinguished party $P^* \in \mathcal{P}$, and integers ℓ, $1 \leq \ell \leq \kappa$, and K, $0 \leq K \leq \lvert\mathcal{V}\rvert$.
Question:	Is there a successful campaign, that is, is it possible to make P^* receive at least ℓ seats using apportionment method $\mathcal{R}$ by changing at most K votes in $\mathcal{V}$?

Note that since $\lvert\mathcal{V}\rvert \geq \kappa \geq \ell$ and $\lvert\mathcal{V}\rvert \geq K$ the encoding of κ, ℓ, and K does not matter for the complexity analysis. $\mathcal{R}$-DESTRUCTIVE-THRESHOLD-APPORTIONMENT-BRIBERY is defined analogously. This time, however, we ask whether it is possible that by changing at most K votes P^* receives *at most* ℓ seats, i.e., we want to limit the parliamentary influence of our target party. In both the constructive and the destructive cases, we assume tie-breaking to be to the advantage of P^*. That is, if P^* and another party P have the same value in their lists and only one seat is left for them, P^* will receive it.

3 Classical Top-Choice Mode

We start by analyzing the complexity of the two problems just defined in the classical top-choice mode of apportionment. We will see that for all divisor sequence methods both deciding whether a successful campaign exists and, if so, computing such a campaign can be done in polynomial time, in both the constructive and the destructive case.

Theorem 1. *Let $\mathcal{R}$ be a divisor sequence method. Then $\mathcal{R}$-*THRESHOLD-APPORTIONMENT-BRIBERY *and its destructive variant are in* P.

The proof of Theorem 1 is presented in the remainder of this section and relies on the lemmas we will present now. Algorithm 1 for $\mathcal{R}$-THRESHOLD-APPORTIONMENT-BRIBERY is given explicitly; we later describe how to adapt it for the destructive case. Note that both algorithms can also easily be adapted to compute an actual campaign (if one exists). The following lemma is crucial for the correctness of the algorithms.

Lemma 1. *The following two statements hold for all divisor sequence methods.*

1. *The maximum number of additional seats for party P^* by bribing at most K votes can always be achieved by convincing exactly K voters from parties in $\mathcal{P}_{-P^*}$ to vote for P^* instead.*
2. *The maximum number of seats we can remove from party P^* by bribing at most K votes can always be achieved by convincing exactly K voters from P^* to vote for parties in $\mathcal{P}_{-P^*}$ instead.*

Proof. Note that when a party receives additional support, the parties fraction list values increase, while they decrease when the support is decreased.

We begin with the first claim. Let's assume we found a way to convice voters to change their vote (in the following called *a bribery action*) such that P^* receives X additional seats. *Case 1:* There are parties other than P^* which receive additional votes. Let $P_i \neq P^*$ be such a party. Now consider that we move all votes that were moved to P_i to P^* instead. Clearly, P^*'s fraction list values increase, while those of P_i decrease. Thus P^* receives at least as many seats in this modified bribery action as in the original. *Case 2:* Votes are only moved from parties in $\mathcal{P}_{-P^*}$ to P^*. By moving (some) voters to party P_i instead of to P^*, the fraction list values of P^* would decrease, while those of P_i would increase. Therefore, P^* cannot receive more seats in this alternative bribery action compared to the original one. Finally, note that by the monotonicity of divisor sequence methods, moving more voters to P^* never makes P^* lose any seats. Thus we can spend the whole budget K on moving voters to P^*. This together with the two given cases implies that the best possible number of additional seats can always be achieved by moving K voters only from $\mathcal{P}_{-P^*}$ to P^* (although there might be other solutions which are equally good).

To prove the second claim, just swap the roles of P^* and the other parties. $\square$

Lemma 1 is crucial for the correctness of the algorithms because it implies that we should exhaust the whole budget K for moving voters from other parties to P^* in the constructive case, and for moving voters from P^* to other parties in the destructive case. That is, we do not need to consider moving voters within $\mathcal{P}_{-P^*}$.

Algorithm 1 decides whether a successful campaign exists in the constructive case. The algorithm works with every divisor sequence method. We now describe the algorithm intuitively. We first set K to the minimum of $n - \sigma(P^*)$ and K because this is the maximum number of votes we can move from other parties to P^*. Should it be impossible for P^* with this K to reach the threshold, we can already answer NO, as P^* never receives any seat at all. The crucial part of the algorithm is computing the γ dictionary: As commented, $\gamma[P][x]$ gives the minimum number of votes that must be removed from party P so that P receives only x seats before P^* receives the ℓ-th seat, assuming P^* has exactly K additional votes in the end. Note that γ can be efficiently

Algorithm 1 Deciding Threshold-Apportionment-Bribery

```
Input: 𝒫, 𝒱, τ, κ, P*, K, ℓ
K ← min{n − σ(P*), K}
if σ(P*) + K < τ then
    return No
end if
compute γ {γ[P][x] is the minimum bribery budget
    needed to ensure that P receives exactly x seats before
    P* gets ℓ seats}
initialize table tab with κ − ℓ columns and m rows,
    where tab[0][0] ← 0 and the other entries are ∞.
let o : {1, ..., |𝒫_{−P*}|} → 𝒫_{−P*} be an ordering

for i ← 1 to |𝒫_{−P*}| do
    for s ← 0 to κ − ℓ do
        for (x, cost) ∈ γ[o(i)] do
            if s − x ≥ 0 then
                tmp ← tab[i − 1][s − x] + cost
                if tmp < tab[i][s] then
                    tab[i][s] = tmp
                end if
            end if
        end for
    end for
end for

for s ← 0 to κ − ℓ do
    if tab[|𝒫_{−P*}|][s] ≤ K then
        return Yes
    end if
end for
return No
```

computed. We describe this in detail later. Note that we now define an order o over the parties. This can be any order; we just use it to identify each party with a row in the table which we now begin to fill. For each i, $1 \leq i \leq |\mathcal{P}_{-P^*}|$, and each s, $0 \leq s \leq \kappa - \ell$, the cell $\mathrm{tab}[i][s]$ contains the minimum number of votes needed to be moved away from parties $o(1), \dots, o(i)$ such that $o(1), \dots, o(i)$ receive s seats in total before P^* is assigned its ℓ-th seat (again, assuming P^* has exactly K additional votes in the end). This table can also be efficiently computed with dynamic programming, as we describe later. Finally, we check if there exists a value of at most K in the last row of the table. If this holds, we answer YES because there do exist bribes that do not exceed K and ensure that the other parties leave the ℓ-th seat for P^*.

Note that by tracing back through the table tab we can find the individual numbers of voters we need to move from each party to P^* for a successful campaign. This number does not sum up to K in many cases. If so, we can simply remove the remaining votes from arbitrary parties (except P^*).

Lemma 2. *Algorithm 1 decides $\mathcal{R}$-THRESHOLD-APPORTIONMENT-BRIBERY for every divisor sequence method $\mathcal{R}$ in polynomial time.*

Proof. We first prove that the algorithm indeed runs in polynomial time. For most of the algorithm this is easy to see: We essentially fill a table with $|\mathcal{P}|$ rows and at most κ columns. Since $\kappa \leq |\mathcal{V}|$, the table size is indeed polynomial in the input size. However, it is yet unclear how γ is computed. Computing γ works with a binary search for the jumping points of a function ϕ. Thereby, ϕ is defined as the number of seats a party with y votes receives before P^* receives ℓ seats, assuming P^* has exactly K additional votes in the end. Let q be the final vote count of P^* (i.e., with the K additional votes). Then, for a divisor sequence method with the sequence $d = (d_1, d_2, \dots, d_\kappa)$ we have

$$\phi(y) = \begin{cases} 0 & \text{if } y \leq \tau \\ 0 & \text{if } y \leq q/d_\ell \\ \max\{z \in \{1, \dots, \kappa\} \mid y/d_z > q/d_\ell\} & \text{otherwise} \end{cases}.$$

Finding the jumping points with binary search is in $\mathcal{O}(\kappa \cdot \log(K))$.

We now prove the correctness of the algorithm. Starting in the beginning, setting K to the minimum of $n - \sigma(P^*)$ and K is necessary to ensure that we never move more voters from parties in $\mathcal{P}_{-P^*}$ than allowed. Setting K higher than that would result in false positive results. For the remainder of this proof, we assume all K votes are moved from $\mathcal{P}_{-P^*}$ to P^*, i.e., P^* receives K additional votes in the end. This is optimal according to Lemma 1. The first if-statement returns NO if P^* cannot reach the threshold. This answer is correct since P^* can never get any seat as long as it is below the threshold, i.e., in this case the bribe is unsuccessful.

In the middle part of the algorithm, we fill a table. Recall that for each i, $1 \leq i \leq |\mathcal{P}_{-P^*}|$, and each s, $0 \leq s \leq \kappa - \ell$, the cell $\mathrm{tab}[i][s]$ contains the minimum number of votes needed to be removed from parties $o(1), \dots, o(i)$ such that $o(1), \dots, o(i)$ receive s seats in total before P^* is assigned its ℓ-th seat. The values are computed dynamically from the previous row to the next row. This is possible because the seats that parties $o(1), \dots, o(i)$ receive in total before P^* is assigned its ℓ-th seat are exactly the sum of the number of seats the parties receive individually before P^* receives its ℓ-th seat. Further, since this number can be computed directly by comparing the divisor list of the party with the divisor list of P^* (i.e., the ϕ function of each party is independent of other parties' support) the required bribery budget is also exactly the sum of the individual bribes. Thus the values in the list are indeed computed correctly.

Finally, if in the last row there exists a value of at most K, we correctly answer YES, by the following argument. Suppose we have a value of at most K in cell $\mathrm{tab}[|\mathcal{P}_{-P^*}|][s]$. Then there are bribes that do not exceed K and ensure that the other parties receive at most s seats before P^* is assigned its ℓ-th seat. Since there are a total of κ seats available, and the other parties get $s \leq \kappa - \ell$ seats before P^* receives the ℓ-th seat, P^* will indeed receive its ℓ-th seat. However, if all cells of the last row contain a value greater than K, the given budget is too small to ensure that the other parties receive at most $\kappa - \ell$ seats before P^* receives its ℓ-th seat. Thus the other parties receive at least $\kappa - \ell + 1$ seats in this case, which leaves at most $\ell - 1$ seats for P^*, so we correctly answer NO. ☐

We can easily adapt Algorithm 1 for the destructive case. This time, we *remove* $\min\{K, \sigma(P^*)\}$ voters from party P^* and *add them to the other* parties. Of course, when P^* is pushed below the threshold, we immediately answer YES. For the destructive case, ϕ and γ need to be defined slightly different. Here, we define $\phi(y)$ as the number of seats a party with y votes receives before P^* is assigned its $(\ell+1)$-th seat (what we try to prevent). And $\gamma[P][x]$ is defined as the minimum number of votes we need to add to party P such that it receives at least x seats before P^* is assigned its $(\ell+1)$-th seat. Again, we fill the table with dynamic programming but this time, whenever we have filled a row completely, we check if it is possible for the parties corresponding to all yet filled rows to receive at least $\kappa - \ell$ seats before P^* receives its $(\ell+1)$-th seat by bribery. That is, we check whether we filled the last cell in the current row with a value of at most K or whether it would be possible to fill a cell beyond the last table cell in the current row with such a value. In that case we answer YES, since there are not enough seats left for P^* to be assigned its $(\ell+1)$-th seat with this bribery action. If this was never possible, we answer NO because the best we could do is to occupy at most $\kappa - \ell - 1$ seats with parties in $\mathcal{P}_{-P^*}$, which still leaves the $(\ell+1)$-th seat for P^*. This sketches the proof of the following lemma and completes the proof of Theorem 1.

Lemma 3. *For every divisor sequence method $\mathcal{R}$, $\mathcal{R}$-DESTRUCTIVE-THRESHOLD-APPORTIONMENT-BRIBERY can be decided in polynomial time.*

Note that since we can decide in polynomial time whether there exists a successful (constructive or destructive) campaign, we can also find the maximum number of seats we can guarantee for P^* with a budget of K (using a simple binary search) in polynomial time. We do this in the experiments of the following section to test the effectiveness of the campaigns on real-world elections.

4 Experiment

As we showed in the previous section, computing successful and even optimal campaigns is computationally tractable. This indicates that such an attack would be relatively simple for a campaign manager to execute (at least from a computational standpoint). An immediate question that arises is whether the campaign is effective enough to be worth to be executed, i.e., how many seats can we actually gain for P^* in an optimal constructive campaign, and how many seats can we take away from P^* in an optimal destructive campaign?

In our experiment, we use three datasets from recent elections shown in Table 1. GREECE2023 is the *Greek parliamentary election 2023* with 300 seats to allocate and a 3% threshold, IKE2022 is the *Israel Knesset election 2022* with 120 seats and a threshold of 3.25%, and BUL2023[3] is the *2023 Bulgarian parliament election* with 240 seats and a 4% threshold. The datasets were taken from the respective Wikipedia sites,[4] with original language data available at https://votes25.bechirot.gov.il/nationalresults, https://ekloges.ypes.gr/current/v/home/parties/, and https://results.cik.bg/ns2023/rezultati/index.html.

GREECE2023	**IKE2022**	**BUL2023**
New Dem. (40.8)	Likud (23.4)	GERB—SDS (26.5)
Syriza (20.1)	Yesh Atid (17.8)	PP–DB (24.6)
PASOK (11.5)	Relig. Zionism (10.8)	Revival (14.2)
Communist (7.2)	National Unity (9.1)	Rights a. Free. (13.8)
Greek Solution (4.5)	Shas (8.3)	BSP (8.9)
Victory (2.9)	United Torah (5.9)	Such a People (4.1)
Freedom (2.9)	Yisrael Beiteinu (4.5)	Bulgarian Rise (3.1)
MeRA25 (2.6)	United Arab List (4.1)	The Left! (2.2)
Subversion (0.9)	Hadash–Ta'al (3.8)	Neutr. Bulgaria (0.4)
National Creat. (0.8)	Labor (3.7)	Together (0.4)

Table 1: The ten largest parties' percentage share in datasets GREECE2023 (*Greek parliamentary election 2023*), IKE2022 (*Israel Knesset election 2022*), and BUL2023 (*Bulgarian parliament election 2023*).

We conducted our experiments as follows. In all three elections, we focus on a budget K equal to 0.25% of the total vote count. This is a relatively small fraction of the voters, and we find it plausible that a campaign manager could be able to influence that many voters. To show the effect of the threshold on the effectiveness of a campaign, we gradually raise the threshold in our experiment. As the distinguished party P^* we always choose the party with the highest voter support in the election, since it reaches all tested thresholds and is thus always present in the parliament. Lastly, we use D'Hondt in our experiments as a representative of the divisor sequence methods, since it is one of the most widely used in apportionment elections. We also conducted the same experiments with Sainte-Laguеë with similar results.

Figure 1 illustrates the effectiveness of both the constructive (top row) and destructive (bottom row) campaigns run on the three real-world elections. One would expect some kind of proportionality, e.g., that 0.25% of the voters control approximately 0.25% of the seats. This is indeed what we observe for many values of the threshold. However, there are some spikes where with only 0.25% of voters one can make P^* gain sometimes 5% or even 10% of all seats on top in the constructive case. This is considerably more that one can expect from our small budget. Note that the spikes always occur at thresholds where a party is directly above the threshold. For instance, in BUL2023 we see a peak at thresholds 2.0%, 2.9%, 3.9%, and 8.7%, which are exactly the values where *The Left!*, *Bulgarian Rise*, *Such a People*, and *BSP* are slightly above the threshold (see Table 1). This indicates that at these thresholds the campaign is focused on pushing a party below the threshold and free up its seats. For the destructive case, we can see similar peaks as in the constructive campaigns. However, this time the peaks are at thresholds where a party is just below the threshold. That is, the campaign is focused on raising a party above the threshold to steal seats from P^*.

Note that we also ran the experiments for other values for

[3]Here, we removed votes from the dataset which are labeled 'none of the above.'

[4]https://en.wikipedia.org/wiki/May_2023_Greek_legislative_election, https://en.wikipedia.org/wiki/2022_Israeli_legislative_election, https://en.wikipedia.org/wiki/2023_Bulgarian_parliamentary_election.

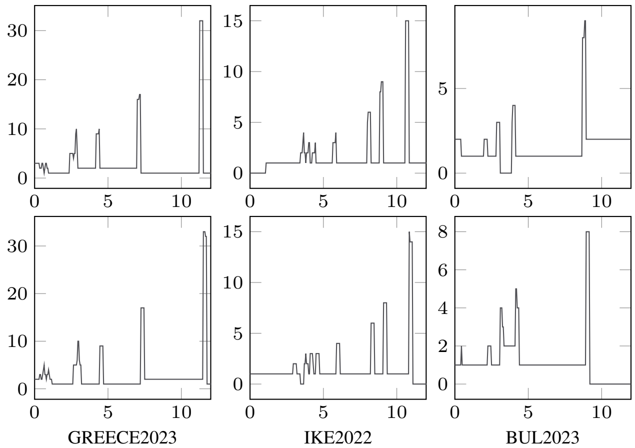


Figure 1: The x-axis shows a variety of thresholds in percent of the number of voters n. The y-axis shows the maximally achievable number of additional seats (prevented seats) for the strongest party by D'HONDT-THRESHOLD-APPORTIONMENT-BRIBERY in the top row and by D'HONDT-DESTRUCTIVE-THRESHOLD-APPORTIONMENT-BRIBERY in the bottom row, each with a given budget of $K = 0.0025 \cdot n$.

the bribe budget K and observed the following: For smaller budgets, we see narrower (and sometimes lower) peaks right at the thresholds where a party is just above it (respectively, just below it, in the destructive case), while for larger budgets, the peaks become wider (and sometimes also higher). Figure 2a shows the results for 0.15% and Figure 2b for 0.35%. Figures 3a and 3b also illustrate the effects when we choose the second- or third-strongest party instead of the strongest as our distinguished party P^*. Again, the results are similar. Only with the third-strongest party do we see a large spike when it is just below (or just above, in the destructive case) the threshold, which, however, is in line with what one would expect.

5 The Second-Chance Mode

From the previous section we know that it is quite problematic if optimal campaigns are easy to compute, because it makes it very simple for a campaign manager to exert an enormous influence on the election outcome. Therefore, it would be of great advantage if there was a modification to the usual apportionment setting that makes the computation of optimal campaigns intractable. As mentioned in the introduction, another general problem of apportionment elections with thresholds is that voters for parties below the threshold are completely ignored. As a result, the parliament tends to be less representative. We now introduce the *second-chance mode* of voting in apportionment elections which will help resolve both of these problems at once. Unlike in the top-choice mode, in the second-chance mode voters for parties below the threshold get a second chance to vote. That is, we first determine the parties $\widehat{\mathcal{P}_\tau}$ that have at least τ top choices, i.e., the parties that make it above the threshold. Each voter now counts as a supporter for their most preferred party in $\widehat{\mathcal{P}_\tau}$. The second-chance voting process is reminiscent of the *single transferable vote (STV)* rule. However, it differs as STV is a single- or multiwinner voting rule and is not used for computing support allocations.

Note that similar voting systems are already being used in Australia e.g. for the *House of Representative* and *Senate*. In those elections, voters rank the candidates or parties from most to least preferred and votes for excluded choices are transferred according to the given ranking until the vote counts. In Section 3, we showed that in the classical apportionment setting, bribery can be solved efficiently for each divisor sequence method. To show that these problems are NP-hard in the second-chance mode, we provide reductions from the NP-complete HITTING SET problem (Karp 1972).

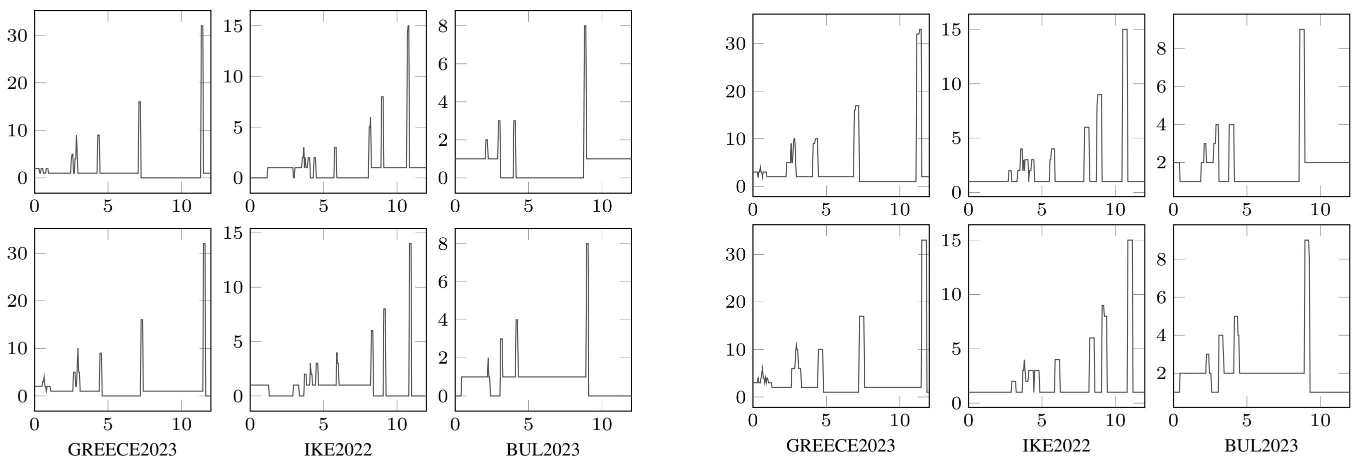
HITTING SET	
Given:	A set $U = \{u_1, \dots, u_p\}$, a collection $S = \{S_1, \dots, S_q\}$ of nonempty subsets of U, and an integer K, $1 \le K \le \min\{p, q\}$.
Question:	Is there a set $U' \subseteq U$, $\lvert U'\rvert \le K$ such that $U' \cap S_i \neq \emptyset$ for each $S_i \in S$?

Instead of just focusing on specific apportionment methods, in the following we generalize our results to a whole class of apportionment methods. We call an apportionment method *majority-consistent* if no party in $\mathcal{P}$ with less support than A receives more seats than A, where $A \in \mathcal{P}$ is a party with the highest support. Undoubtedly, this is a criterion every reasonable apportionment method should satisfy. Note that all divisor sequence methods and also the common Largest-Remainder-Method (LRM) (see, e.g., Bredereck et al. (2020)) are majority-consistent. We now show that the second-chance mode of apportionment voting makes computing an optimal strategic campaign computationally intractable, and can prevent attempts of running them.

Theorem 2. *For each majority-consistent apportionment method $\mathcal{R}$, $\mathcal{R}$-THRESHOLD-APPORTIONMENT-BRIBERY and $\mathcal{R}$-DESTRUCTIVE-THRESHOLD-APPORTIONMENT-BRIBERY are* NP*-hard in the second-chance mode. They are* NP*-complete if $\mathcal{R}$ is polynomial-time computable.*

Proof. Membership of both problems in NP is obvious whenever $\mathcal{R}$ is polynomial-time computable. We show NP-hardness of $\mathcal{R}$-THRESHOLD-APPORTIONMENT-BRIBERY by a reduction from HITTING SET. Let $(U, S, K) = (\{u_1, \dots, u_p\}, \{S_1, \dots, S_q\}, K)$ be an instance of HITTING SET with $q \ge 4$. In polynomial time, we construct an instance of $\mathcal{R}$-THRESHOLD-APPORTIONMENT-BRIBERY with parties $\mathcal{P} = \{c, c'\} \cup U$, a threshold $\tau = 2q + 1$, $\ell = 1$ desired seat, $\kappa = 1$ available seat, and the votes

$$\begin{aligned}
\mathcal{V} = (4q + 2 \text{ votes} \quad & c \succ \cdots, \\
4q + K + 2 \text{ votes} \quad & c' \succ \cdots, && (1)\\
\text{for each } j \in [q], 2 \text{ votes} \quad & S_j \succ c' \succ \cdots, && (2)\\
\text{for each } i \in [p], q - \gamma_i \text{ votes} \quad & u_i \succ c \succ \cdots, && (3)\\
\text{for each } i \in [p], q - \gamma_i \text{ votes} \quad & u_i \succ c' \succ \cdots), && (4)
\end{aligned}$$



(a) Experiment with a budget of $K = 0.0015 \cdot n$.

(b) Experiment with a budget of $K = 0.0035 \cdot n$.

Figure 2: The x-axis shows a variety of thresholds in percent of the number of voters n. The y-axis shows the maximally achievable number of additional seats (prevented seats) for the strongest party by D'HONDT-THRESHOLD-APPORTIONMENT-BRIBERY in the top row and by D'HONDT-DESTRUCTIVE-THRESHOLD-APPORTIONMENT-BRIBERY in the bottom row.

where $S_j \succ c'$ denotes that each element in S_j is preferred to c', but we do not care about the exact order of the elements in S_j. Further, $2\gamma_i$ is the number of votes from group (2), in which u_i is at the first position. That is, it is guaranteed that each u_i receives exactly $2\gamma_i + (q - \gamma_i) + (q - \gamma_i) = 2q < \tau$ top choices, while c has $4q+2 \geq \tau$ and c' has $4q+K+2 \geq \tau$ top choices. Note that the voters in groups (2) and (4) use their second chance to vote for c', and those in group (3) use it to vote for c. It follows that c' currently receives exactly $2q + K$ more votes than c and thus wins the seat. We now show that we can make the distinguished party $P^* = c$ win the seat by bribing at most K voters if and only if there is a hitting set of size at most K.

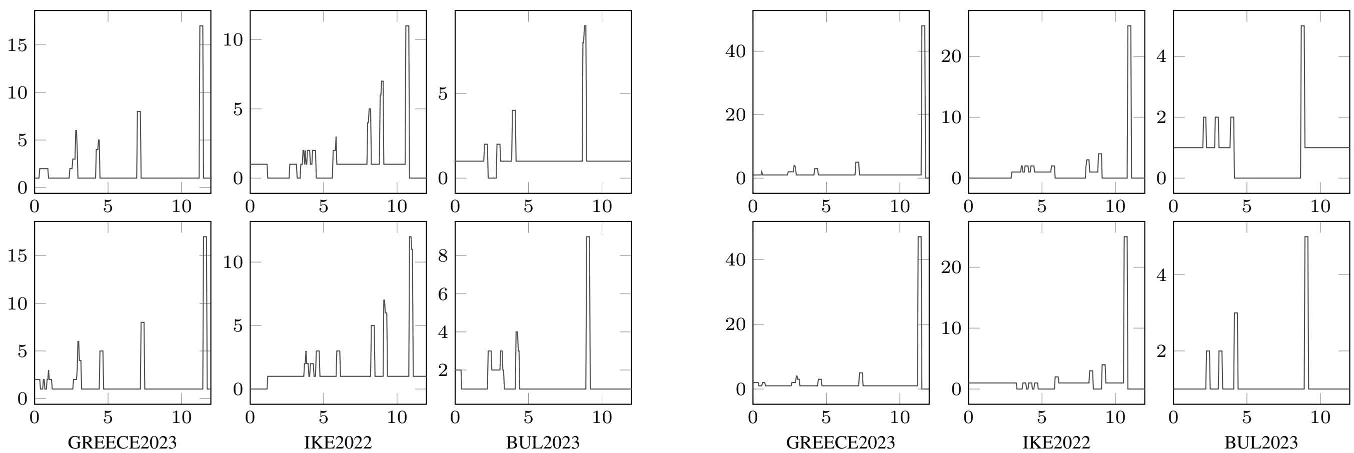
($\Leftarrow$) Suppose there exists a hitting set $U' \subseteq U$ of size exactly K (if $|U'| < K$, it can be padded to size exactly K by adding arbitrary elements from U). For each $u_i \in U'$, we bribe one voter from group (1) to put u_i at their first position. These u_i now each receive the $2q + 1$ top choices required by the threshold, i.e., they participate in the further apportionment process. Each u_i can receive a support of at most $4q + 1$. Since the support of c is not affected by any bribes, no u_i can win the seat against c. Groups (3) and (4) do not change the support difference between c and c' and thus can be ignored. However, since U' is a hitting set, all $2q$ voters in group (2) now vote for a party in U' instead of c', reducing the difference between c and c' by $2q$. Further, we have bribed K voters from group (1) to not vote for c', which reduces the difference between c and c' by another K votes. Therefore, c and c' now have the same support, and since we assume tie-breaking to prefer c, party c wins the seat.

($\Rightarrow$) Suppose the smallest hitting set has size $K' > K$. That is, with only K elements of U we can hit at most $q - 1$ sets from S. It follows that by bribing K voters from group (1) to vote for some $u_i \in U$ instead of c', we can only prevent up to $2(q - 1)$ voters from group (2) to use their second chance to vote for c'. Thus we reduce the difference between c and c' by at most $2(q - 1) + K$, which is not enough to make c win the seat. Now consider that we do not use the complete budget K on this strategy, i.e., to bribe voters of group (1), but only $K'' < K$. Note that by bringing only K'' parties from U above the threshold, we can only hit up to $2(q - 1 - (K - K''))$ sets from S. So the difference between c and c' is reduced by at most $2(q - 1 - (K - K'')) + K''$ using this strategy. However, we now have a budget of $K - K''$ left to bribe voters, e.g., from group (2), to vote primarily for c without bringing any additional u_i above the threshold. It is easy to see that we will only reduce the difference between c and c' by at most $2(K - K'')$ with this strategy as, in the best case, c gains one supporter and c' loses one with a single bribery action. Thus we cannot reduce the difference between c and c' by more than $2(q-1-(K-K''))+K''+2(K-K'') = 2(q-1)+K''$ with this mixed strategy. For each $K'' \leq K$, we have $2(q - 1) + K'' < 2q + K$. Therefore, if there is no hitting set of size at most K, we cannot make the distinguished party c win against c'.
The proof for the destructive variant works by swapping the roles of c and c'. □

6 Conclusions

We have studied strategic campaigns for apportionment elections with thresholds and introduced the second-chance mode of voting, where voters for parties below the threshold get a second chance to vote. The second-chance mode makes computing strategic campaigns intractable while they are easy to compute in the classical top-choice mode.

As future work, we propose to study other types of strategic campaigns (e.g., cloning of parties; see (Tideman 1987; Elkind, Faliszewski, and Slinko 2011; Neveling and Rothe 2020)). We already studied electoral control problems, in particular constructive and destructive control by adding or deleting parties or votes. It turns out that both, top-choice



(a) Experiment with the second-best party.

(b) Experiment with the third-strongest party.

Figure 3: The x-axis shows a variety of thresholds in percent of the number of voters n. The y-axis shows the maximally achievable number of additional seats (prevented seats) by D'HONDT-THRESHOLD-APPORTIONMENT-BRIBERY in the top row and D'HONDT-DESTRUCTIVE-THRESHOLD-APPORTIONMENT-BRIBERY in the bottom row, each with a given budget of $K = 0.0025 \cdot n$.

and second-chance mode are resistant to all four party control problems. For the proofs, it suffices to adapt the proofs of Bartholdi et al. (Bartholdi III, Tovey, and Trick 1992) and Hemaspaandra et al. (Hemaspaandra, Hemaspaandra, and Rothe 2007) showing that plurality voting is resistant to the corresponding control problems. For vote control in the top-choice mode, Algorithm 1 can be adapted showing that all four cases of vote control are in P for divisor sequence methods. However, this only works when the threshold is fixed, i.e., not given as percent of n. Regarding the second-chance mode, we have so far been able to show that all majority-consistent apportionment methods are resistant to vote control if the threshold is fixed. Another direction for future research is to study the complexity of these problems in restricted domains such as (nearly) single-peaked preferences (Faliszewski, Hemaspaandra, and Hemaspaandra 2011; Faliszewski et al. 2011). Also, studying the effectiveness of strategic campaigns in the second-chance mode using ILPs or approximation algorithms is an interesting direction for the future.

To make our strategic campaigns even more realistic, we propose to study more sophisticated cost functions such as *distance bribery* (Baumeister, Hogrebe, and Rey 2019) where the cost of bribing a voter depends on how much we change the vote. We conjecture the problem to be harder under the assumption of distance bribery because of the observation that Lemma 1 no longer holds. That is, there are cases where it is more effective to move votes within $\mathcal{P}_{-P^*}$ than to move them to P^*. To illustrate this, suppose we have two seats, $\sigma(P^*) = 7$, $\sigma(P_A) = 4$, and $\sigma(P_B) = 2$ with $\tau < 2$. According to D'Hondt, P^* and P_A each receive one seat. Say $K = 1$ but the cost for changing a vote from P_A to P^* is 2, and the cost for changing a vote from P_B to P^* is even higher. We thus cannot move a single voter to P^*, i.e., we cannot gain any seats for P^* by this strategy. However, if the cost for moving a voter from P_A to P_B is 1, we gain one seat for P^* by moving a voter from P_A to P_B.

While NP-hardness is desirable in the context of strategic campaigns, in the context of, e.g., margin of victory or robustness, the interpretations can be flipped, which can also be studied as future work. Finally, we suggest studying the extent to which voters' satisfaction with the parliament increases when the second-chance mode is used.

Acknowledgements This work was supported in part by DFG grant RO-1202/21-1. We thank Niclas Boehmer, Robert Bredereck, and Martin Bullinger for their helpful comments during a seminar at Schloss Dagstuhl.

References

Balinski, M.; and Young, H. 1975. The Quota Method of Apportionment. *The American Mathematical Monthly*, 82(7): 701–730.

Balinski, M.; and Young, H. 1982. *Fair Representation: Meeting the Ideal of One Man, One Vote*. New Haven: Yale University Press.

Bartholdi III, J.; Tovey, C.; and Trick, M. 1992. How Hard Is It to Control an Election? *Mathematical and Computer Modelling*, 16(8/9): 27–40.

Baumeister, D.; Hogrebe, T.; and Rey, L. 2019. Generalized Distance Bribery. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 1764–1771. AAAI Press.

Baumeister, D.; and Rothe, J. 2015. Preference Aggregation by Voting. In Rothe, J., ed., *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Springer Texts in Business and Economics, chapter 4, 197–325. Springer-Verlag.

Bredereck, R.; Faliszewski, P.; Furdyna, M.; Kaczmarczyk, A.; and Lackner, M. 2020. Strategic Campaign Management in Apportionment Elections. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 103–109. ijcai.org.

Elkind, E.; Faliszewski, P.; and Slinko, A. 2011. Cloning in Elections: Finding the Possible Winners. *Journal of Artificial Intelligence Research*, 42: 529–573.

Faliszewski, P.; Hemaspaandra, E.; and Hemaspaandra, L. 2009. How Hard Is Bribery in Elections? *Journal of Artificial Intelligence Research*, 35: 485–532.

Faliszewski, P.; Hemaspaandra, E.; and Hemaspaandra, L. 2011. The Complexity of Manipulative Attacks in Nearly Single-Peaked Electorates. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge*, 228–237. ACM Press.

Faliszewski, P.; Hemaspaandra, E.; Hemaspaandra, L.; and Rothe, J. 2011. The Shield that Never Was: Societies with Single-Peaked Preferences are More Open to Manipulation and Control. *Information and Computation*, 209(2): 89–107.

Faliszewski, P.; and Rothe, J. 2016. Control and Bribery in Voting. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A., eds., *Handbook of Computational Social Choice*, chapter 7, 146–168. Cambridge University Press.

Hemaspaandra, E.; Hemaspaandra, L.; and Rothe, J. 2007. Anyone But Him: The Complexity of Precluding an Alternative. *Artificial Intelligence*, 171(5–6): 255–285.

Karp, R. 1972. Reducibility among Combinatorial Problems. In Miller, R.; and Thatcher, J., eds., *Complexity of Computer Computations*, 85–103. Plenum Press.

Neveling, M.; and Rothe, J. 2020. The Complexity of Cloning Candidates in Multiwinner Elections. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 922–930. IFAAMAS.

Oelbermann, K.-F.; and Pukelsheim, F. 2020. The European Elections of May 2019: Electoral Systems and Outcomes. Study for the European Parliamentary Research Service.

Pellicer, M.; and Wegner, E. 2014. The mechanical and psychological effects of legal thresholds. *Electoral Studies*, 33: 258–266.

Pukelsheim, F. 2017. *Proportional Representation*. Springer.

Tideman, N. 1987. Independence of Clones as a Criterion for Voting Rules. *Social Choice and Welfare*, 4(3): 185–206.