CrossMark

Computational aspects of strategic behaviour in elections with top-truncated ballots

Vijay Menon[1] · **Kate Larson**[1]

Abstract Understanding when and how computational complexity can be used to protect elections against different manipulative actions has been a highly active research area over the past two decades. Much of this literature, however, makes the assumption that the voters or agents specify a complete preference ordering over the set of candidates. There are many multiagent systems applications, and even real-world elections, where this assumption is not warranted, and this in turn raises a series of questions on the impact of partial voting on the complexity of manipulative actions. In this paper, we focus on two of these questions. First, we address the question of how hard it is to manipulate elections when the agents specify only top-truncated ballots. Here, in particular, we look at the weighted manipulation problem—both constructive and destructive manipulation—when the voters are allowed to specify top-truncated ballots, and we provide general results for all scoring rules, for elimination versions of all scoring rules, for the plurality with runoff rule, for a family of election systems known as Copeland$^\alpha$, and for the maximin protocol. The second question we address is the impact of top-truncated voting on the complexity of manipulative actions in electorates with structured preference profiles. In particular, we consider electorates that are single-peaked and we show how, for many voting protocols, allowing top-truncated voting reimposes the $\mathcal{NP}$-hardness shields that normally vanish in such electorates.

Keywords Computational social choice · Elections · Complexity of manipulative actions · Top-truncated voting · Manipulation · Bribery

---

✉ Vijay Menon
  vijay.menon@uwaterloo.ca

  Kate Larson
  kate.larson@uwaterloo.ca

[1] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada

1 Introduction

Preference aggregation is an important problem in multiagent settings as there are many scenarios where a group of agents has to make a common decision. The process of arriving at this decision, in turn, has to accommodate the needs and preferences of all the participating agents. A natural, and commonly used, mechanism to achieve this is voting, where all the agents specify their preferences and a previously agreed-upon procedure—called the election system or voting protocol—is used to arrive at the decision. Election problems abound in human contexts. From electing a government to hiring within a CS department to choosing a winner in television reality shows, they are everywhere. Such problems are increasingly ubiquitous in multiagent settings. For instance, it has been proposed as a mechanism for web spam reduction [19], for collaborative filtering and recommender systems [45], and for multiagent planning [21].

Given the fact that it is useful and a widely used mechanism, voting, naturally, has been extensively studied and it has been found that it is not without its problems. In fact, the theory of social choice, which focuses on the study of collective decision making processes and procedures, has a number of impossibility results surrounding fundamental issues that arise in running elections. Among many such issues, one natural worry that can arise is "What if the agents try to manipulate the outcome of an election by, say, misreporting their true preferences?". From the perspective of designing voting systems, this question has been answered through the Gibbard–Satterthwaite Theorem [37,46] and its generalizations (like the Duggan–Schwartz Theorem [18]) which essentially say that there are no reasonable voting systems which aren't manipulable. However, despite such results, there still remains the question of how to achieve these manipulations. Starting with the seminal work of Bartholdi, Orlin, Tovey, and Trick through a series of papers [3–5], there has been much work that has raised this question and has looked into how computational complexity can be used as a barrier to protect elections against different manipulative actions (see [27,28] for fairly recent surveys and also [14, chap. 6–7]). This paper too tries to address the question of when computational complexity can be used to protect elections, and in particular we consider the case when the agents are allowed to submit truncated ballots—meaning that they are not obligated to specify their ranking of all the candidates but need to do so only for a subset of them.

Now, the first question that could arise here is: "why truncated ballots?"; after all, there has been a flurry of papers in computational social choice that have studied the complexity of manipulative actions under the assumption that the agents submit complete rankings over the set of candidates. There are a number of reasons why looking at partial preferences in general is important. First, expecting agents to provide complete ballots is only practical when the number of candidates is small and in practice there are many situations where this might not be the case. As an instance, consider the problem of building a meta-search engine as outlined by Dwork et al. [19] by combining search results from individual search engines. In this case, for a given query, if the search results of a particular search engine denotes its "vote", and the candidates are all the possible search results, then it is easy to see that the votes have to be necessarily truncated before combining. Second, even if the number of candidates were not very large, agents might just be unable to specify a complete ordering because they do not have enough information on them. Third, there are many real-world elections in which voters are allowed to submit partial orderings. For example, apart from the fact that most political elections use the plurality rule where every agent only specifies their most favourite candidate, the elections for the President of Ireland and for the legislative assembly in the Australian Capital Territory uses the Single Transferable Vote where an agent

can rank a subset of the candidates. Thus, it is important to understand the repercussions of having partially specified preferences ("partial votes") in general. While a "partial vote" can refer to any partial ordering over the set of candidates, here we focus on one kind of "partial vote", namely top-truncated votes. Top-truncated votes are natural in many settings where an agent is certain about its most preferred candidates but is indifferent among the remaining ones or is unsure about them.

In particular, this paper addresses the following questions about elections with truncated ballots.

– While, as outlined above, allowing voters to submit top-truncated ballots can be extremely useful, one of the foremost questions that arises here is: "What happens to the complexity of manipulation if the agents reveal only top-truncated preference orderings?". It is this question that we try to address first, in Sect. 4. Specifically, we study the weighted manipulation problem—both constructive and destructive manipulation—when the voters are allowed to specify top-truncated ordering and we provide many general results. Subsequently, we remove the assumption that the manipulators have complete information and instead look at the impact on complexity of manipulation when there is uncertainty about the non-manipulators' votes.

– There are many elections scenarios where the agents have structured preferences— meaning that there are some valid votes that are never cast. We consider one of the most widely studied such structured preferences, namely single-peaked preferences, and we ask the following question: "What happens to the complexity of manipulative actions when agents have structured preferences and are allowed to submit top-truncated ballots?". In particular, in Sect. 5, we look at the problem of weighted manipulation and weighted bribery in single-peaked electorates, and we show a number of surprising results regarding the impact of top-truncated ballots on the complexity of these manipulative actions in such electorates.

In addressing the above mentioned questions, this paper makes the following contributions:

1. We (in Sects. 4.1 and 4.2) provide general results for all scoring rules, for elimination versions of all scoring rules, for the plurality with runoff rule, for a family of election systems known as Copeland$^\alpha$, and for the maximin protocol, thus extending the preliminary work of Narodytska and Walsh [42] on the impact of top-truncated voting on complexity of manipulation. Subsequently, in Sect. 4.3, we look at the impact on complexity of manipulation when there is uncertainty about the non-manipulators' votes and when the voters are allowed to submit top-truncated ballots.

2. We systematically study the impact of partial voting on manipulative actions in structured preference profiles. Here (in Sects. 5.2 and 5.3) we look at the problem of manipulation and bribery in single-peaked settings when top-truncated ballots are allowed and show a number of surprising results where allowing top-truncated voting seems to reinstate combinatorial protections (that normally vanish when voters are only allowed to present complete orders) for many voting rules against manipulation and bribery. In particular, under the assumption that the voters submit complete ballots, we first provide polynomial time algorithms for manipulation and weighted-bribery for certain voting rules in single-peaked settings, thus extending the works of Faliszewski et al. [29] for manipulation and Brandt et al. [9] for bribery. We then show how these polynomial-time problems become $\mathcal{NP}$-complete when top-truncated ballots are allowed.

3. Finally, we show an example of a natural voting rule where, contrary to intuition, the complexity of manipulation actually increases when moving from the general case (i.e., when there is no restriction on the preferences) to the single-peaked case. In particular,

in Theorem 31 we show how the complexity of manipulating eliminate(Veto), when top-truncated ballots are allowed, moves from being in $\mathcal{P}$ in the general case to being $\mathcal{NP}$-complete in the single-peaked case.

2 Related work

The study of computational complexity for manipulative actions was started through a series of papers by Bartholdi, Orlin, Tovey, and Trick [3–5]. Subsequently, there has been a flurry of work in this domain of research which has studied the computational complexity of manipulative actions in many different settings. The reader can refer to some fairly recent surveys [27,28] and also the book chapters [14, chap. 6–7] for discussions on the evolution of this domain.

Broadly, there are two lines of research that are closely related to our work. First is the work on election problems when the agents are allowed to specify partial preferences. Among them, the papers that are most related to our results here are those of Narodytska and Walsh [42], and Fitzsimmons and Hemaspaandra [33]. Narodytska and Walsh [42] were the first to look at complexity of constructive manipulation under top-truncated voting and they provided an analysis for weighted and unweighted elections using three particular voting protocols, namely Borda, STV, and the Copeland rule. In fact, results in Sect. 4.1 of this paper can be considered as a direct extension to their work. The other most relevant work is that of Fitzsimmons and Hemaspaandra [33] where they looked into how the complexity of bribery, control, and manipulation is affected when different types of partial preferences are allowed. Here we note except for one theorem (Theorem 10)—which was proved independently by Fitzsimmons and Hemaspaandra [33, Theorem 14]—and the case of the Borda scoring rule in our Theorems 27 and 28 (which were shown by Fitzsimmons and Hemaspaandra [33, Theorems 4, 5]), none of our other results overlap with theirs as all their results are derived using only one of the following protocols: Borda, plurality, t-approval, and Copeland$^{\alpha}$. While most of the results in both of these papers and our results in Sect. 4 assumes the general preference setting (i.e., one where there is no restriction on the structure of the agents' preferences), Sect. 5 of this paper looks at the complexity of manipulation and bribery with top-truncated ballots when the preferences of the agents have some structure. Besides the two papers mentioned above, we also note that, subsequent to the conference version of our results in Sect. 5, there is very recent work by Fitzsimmons and Hemaspaandra [34] where they look at the complexity of manipulation with different types of partial preferences in the single-plateaued setting and for a single-peaked setting in which at most two ties are allowed per position.

Additionally, there has also been some work in this line of research that has looked at other election problems when preferences are only partially specified. Among these, the work of Konczak and Lang [39] introduced the possible and necessary winners problem, and Xia and Conitzer [48] extended this further to study the possible and necessary winners problem for many different voting rules when the number of candidates are unbounded and the elections unweighted. Furthermore, Lu and Boutilier [41] looked at the multi-winner problem when only partial preferences are provided, and Baumeister et al. [6] discussed planning various kinds of campaigns in settings where the ballots can be truncated at the top, bottom, or both. A special case of the extension-bribery problem they introduced in that paper is closely related to the manipulation problem with top-truncated ballots.

The second line of research that is closely related to results here is the work on structured preference profiles. This line of research has mainly looked at single-peaked

preferences and more recently at nearly single-peaked preferences. The notion of single-peaked preferences was introduced by Black [7] and subsequently there has been a lot of work in the social choice literature on the same. Among these, in particular, we note the work of Cantala [11] who introduced the concept of "single-peaked with outside options" which is identical to the notion of single-peaked with top-truncated ballots that we study here in Sect. 5, and the work of Barberá [1] who discussed how properties of different variants of single-peaked preferences change for varying amounts of indifference permitted.

In computational social choice, three papers that are in this line of research and are most related to our work are those of Faliszewski et al. [29], Brandt et al. [9], and Faliszewski et al. [30]. The first two discuss manipulation and control, and bribery, respectively, and show how most of the $\mathcal{NP}$-hardness shields for these manipulative actions vanish in single-peaked settings. The third paper studies the complexity of manipulative actions in nearly single-peaked electorates and shows how in many cases the hardness results evaporate. Our work in Sect. 5, in contrast to all the ones above, follows the theme of reinstating these combinatorial protections by allowing top-truncated voting.

Finally, although the above mentioned papers are by far the most related to our work, it is worth noting that one problem which has received considerable attention from researchers in computational social choice is that of single-peaked consistency where, informally, the task is to determine if a given set of preferences is single-peaked or otherwise (see [2,17,23,24, 32,40]).

3 Preliminaries

We model an election as a pair $\mathcal{E} = (C, V)$, where $C = \{c_1, \ldots, c_m\}$ is the set of candidates and $V = \{v_1, \ldots, v_n\}$ is the set of voters. Each voter in turn is modeled as a pair $v_i = (\succ_i, w_i)$, where $\succ_i$ is the preference order of v_i over C and w_i is a positive integer weight associated with the voter v_i. A preference order, $\succ_i$, is said to be a complete order (or a complete vote) when it is antisymmetric, transitive, and a total ordering on C. On the other hand, $\succ_i$ is said to be a top-truncated order (or simply, a top order) when it is a linear order over a non-empty subset of C and where all the unranked candidates are tied and are assumed to be ranked below the ranked candidates. Throughout, a strict preference between two candidates is denoted by ">", while a tie between them is denoted by "$\sim$".

A preference profile is a vector $P = \langle \succ_1, \ldots, \succ_n \rangle$ of individual preferences. An election, $\mathcal{E}$, is said to be unweighted if for all $v_i \in V$, $w_i = 1$. On the other hand, it is said to be weighted if $\exists v_i \in V$ such that $w_i > 1$.

3.1 Voting protocols

An election system or a voting protocol takes in a preference profile, P, as input and it outputs a collection $W \subseteq C$ called the winner set. The following are the commonly-studied voting rules that we consider in this paper. We first present their original definitions which is on complete orders and then discuss how top orders can be handled.

1. *Positional scoring rules* A positional scoring rule is defined by a scoring vector $\alpha = \langle \alpha_1, \ldots, \alpha_m \rangle$, where $\alpha_1 \geq \cdots \geq \alpha_m$. For each voter $v \in V$, a candidate receives α_i points if it is ranked in the ith position by v. In a scoring rule, all the candidates with highest total score form the winner set. Some examples of scoring rules are the *plurality*

rule with $\alpha = \langle 1, \overbrace{0, \ldots, 0}^{m-1} \rangle$, the *Borda rule* with $\alpha = \langle m-1, m-2, \ldots, 0 \rangle$, and the

veto rule with $\alpha = \langle \overbrace{1, \ldots, 1}^{m-1}, 0 \rangle$.

2. *Scoring elimination rules* Let X be any scoring rule. Given a complete ordering, eliminate(X) is the rule that successively eliminates the candidate with the lowest score according to X. Once a candidate is eliminated, the rule is then repeated with the reduced set of candidates (meaning that if there were k candidates previously, after a candidate is eliminated, we treat the instance as one with $k-1$ candidates and apply the rule for these $k-1$ candidates) until there is a single candidate left (in which case the candidate is the unique winner) or all the candidates remaining have the same score (in which case all these candidates form the winning set). The elimination order, e, for an elimination rule is the order in which the candidates are successively eliminated. For instance, $e = (c_1, \ldots, c_k)$ implies that c_1 was the first candidate to be eliminated while c_k was the last. Here, we mainly consider two scoring elimination rules: *eliminate(Borda)*—which is also known as Baldwin's rule or Fishburn's version of Nanson's rule [44]—and *eliminate(Veto)*.

3. *Plurality with runoff* The plurality with runoff rule proceeds in two steps. In the first step, all the candidates except the top two with the most number of first votes are eliminated. This is followed by a second round where the winner is determined by a pairwise comparison between the top two candidates.

4. *Copeland$^\alpha$* Let $\alpha \in \mathbb{Q}$, $0 \le \alpha \le 1$. In Copeland$^\alpha$ (introduced by Copeland [15] for $\alpha = \frac{1}{2}$ and parameterized by Faliszewski et al. [26]), for each pair of candidates, the candidate preferred by the majority receives one point and the other one receives a 0. In case of a tie, both receive α points. Here all the candidates whose total score is at least as high as that of all the other candidates form the winner set.

5. *Maximin* Let $N_P(c_i, c_j)$ denote the number of voters who prefer c_i over c_j in the preference profile, $\mathcal{P}$. Then the maximin score of c_i is $s_i = \min_{j \ne i} N_{\mathcal{P}}(c_i, c_j)$. The winners in the maximin rule are the ones with the highest score.

To deal with top-truncated orders in positional scoring rules where a voter ranks only k out of the m candidates ($k < m$), we use the following three schemes that were used by Narodytska and Walsh [42] in their preliminary work on manipulation with top orders (Emerson [20] also used the same schemes for the Borda rule).

1. *Round-up* A candidate ranked in the ith position ($i \le k$) receives a score of α_i, while all the unranked candidates receive a score of α_m. For any positional scoring rule X, we denote this by $X_\uparrow$. Here we note that the only difference between the Round-up scheme defined by Narodytska and Walsh [42] and the one defined here is with respect to the score given to the unranked candidates. They assign a score of 0 to all the unranked candidates, while we assign them a score of α_m.

2. *Round-down* A candidate ranked in the ith position ($i \le k$) receives a score of $\alpha_{m-(k-i)-1}$, while all the unranked candidates receive a score of α_m. For any positional scoring rule X, we denote this by $X_\downarrow$.

3. *Average score* A candidate ranked in the ith position ($i \le k$) receives a score of α_i, while all the unranked candidates receive a score of $\frac{\sum_{k < j \le m} \alpha_j}{m-k}$. For any positional scoring rule X, we denote this by X_{av}.

Example 1 Let $C = \{c_1, c_2, c_3, c_4\}$ and let v be a voter with top-truncated order $\succ_v$: ($c_3 > c_1 > c_2 \sim c_4$). Then, in the round-up case, c_3 receives a score of α_1, c_1 receives a score of α_2, and both c_2 and c_4 receive α_4. In the round-down case, c_3 receives a score of α_2, c_1

receives a score of α_3, and both c_2 and c_4 receive a score of α_4, and in the average score case, c_3 receives a score of α_1, c_1 receives a score of α_2, and both c_2 and c_4 receive a score of $\frac{\alpha_3+\alpha_4}{2}$.

In scoring elimination rules and plurality with runoff rule, top-truncated votes are dealt with by using the round-up scheme described above. Here, we consider a vote to be valid only until at least one of the candidates listed in it is remaining in the election. In other words, we simply ignore a vote once all the candidates listed in it are eliminated. In the case of Copeland$^\alpha$ and the maximin rule, top orders are dealt with by just keeping to the definition which assumes that all the unranked candidates are tied and are ranked below the ranked candidates.

Condorcet-consistent voting rules An important property of a candidate is that of being a Condorcet winner. A candidate is said to be a Condorcet winner (respectively, a weak-Condorcet winner) if it is preferred over every other candidate by a strict majority of the voters (respectively, by at least half of the voters). While it is not necessary that every election instance has a Condorcet winner or even a weak Condorcet winner, a voting rule which, on every input that has a weak-Condorcet winner, outputs the set of all weak-Condorcet winners as the set of winners is said to be weak-Condorcet consistent. Among the voting rules defined above, the maximin rule [31] and Copeland[1] [9] are examples of weak-Condorcet consistent rules.

3.2 Manipulative actions

We consider two weighted manipulative action problems, namely manipulation and bribery. In particular, we study the Constructive Coalitional Weighted Manipulation (CWCM) problem, the Destructive Coalitional Weighted Manipulation (DWCM) problem, and the Weighted-bribery problem. CWCM and DWCM were first studied by Conitzer et al. [13] and are described below. Informally, the main objective in CWCM is to decide if it is possible for a certain set of voters (manipulators) to vote in such a way that it will result in a preferred candidate winning, while in DWCM it is to decide if it is possible for the manipulators to vote in such a way so as to ensure that a particular disliked candidate will not be a winner.

Definition 1 (*CWCM*) Given a set of candidates C, a set of weighted votes S (preferences of the non-manipulators), the weights for a set of votes T (manipulators' votes), and a preferred candidate p, we are asked if there exists a way to cast the votes in T (i.e., for all $v \in T$, determine their $\succ_v$) so that p is a winner in the election $\mathcal{E} = (C, S \cup T)$.

Definition 2 (*DWCM*) Given a set of weighted votes S (votes of the non-manipulators), the weights for a set of votes T (manipulators' votes), and a disliked candidate h, we are asked if there exists a way to cast the votes in T so that h does not win the election.

The complexity-theoretic study of the bribery problem was first introduced by Faliszewski et al. [25]. The main objective in the bribery problem is to decide if there exists a subset of voters whose votes can be changed, within a certain budget, to make a preferred candidate win. Here we look at the weighted version of the bribery problem which is described below.

Definition 3 (*Weighted-bribery*) Given a set of candidates C, the set of weighted votes V, a preferred candidate p, and a limit $k \in \mathbb{N}$, we are asked if there exists a way to change the votes of at most k of the voters in V so that it results in p being a winner.

While there are two possible models—the unique winner model, where the objective is to make the preferred candidate the unique winner, and the non-unique winner model, where

the objective is to make the preferred candidate *a* winner—under which the problems defined above can be studied, throughout, unless otherwise specified, we use the non-unique winner model as our standard model. Also, we do not discuss how ties are handled since, as stated above, the standard model we use throughout is the non-unique winner model where the objective is just to make the candidate *p* a winner. As for the results which use the unique-winner model, we are essentially assuming that the candidate *p* needs to win without having to resort to any tie-breaking.

3.3 Computational complexity

For most of the $\mathcal{NP}$-hardness results in this paper, we use reductions from either the well-known $\mathcal{NP}$-complete problem Partition (see, e.g., [36]), from a variant of the Partition problem (Partition′), or from a variant of the subset sum problem (Fixed-Difference Subset Sum).

Definition 4 (*Partition*) Given a set of non-negative integers $S = \{a_i\}_{1 \leq i \leq n}$ summing to $2K$, we are asked if there exists a subset S_1 of S which sums to K.

Definition 5 (*Partition'*) Given a set of non-negative integers $S = \{a_i\}_{1 \leq i \leq n}$ such that $\sum_i a_i = 2nK$ and $a_i \geq K$, for $1 \leq i \leq n$, we are asked if there exists a subset S_1 such that $\sum S_1 = nK$, where $\sum S_1$ denotes the sum of all the elements in the set S_1.

The $\mathcal{NP}$-completeness of Partition′ can be shown by a reduction from the Partition problem and the proof can be found in the appendix.

Lemma 1 *Partition′ is NP-complete.*

Definition 6 (*Fixed-difference subset sum*) Given a set of non-negative integers $S = \{a_i\}_{1 \leq i \leq n}$ summing to $2K$, we are asked if there exist two disjoint subsets S_1, S_2 of S such that $\sum S_1 - \sum S_2 = K$, where $\sum S_i$ denotes the sum of all the elements in the set S_i.

The $\mathcal{NP}$-completeness proof of Fixed-Difference Subset Sum is very similar to the one for Partition′ and can be found in the appendix.

Lemma 2 *Fixed-Difference Subset Sum is $\mathcal{NP}$-complete.*

4 Complexity of manipulation in elections with top-truncated ballots

In this section, we address the question of how hard it is to manipulate elections if the agents reveal only top-truncated preference orderings. In particular, we first look at the weighted manipulation problem—both constructive and destructive manipulation—in Sects. 4.1 and 4.2 under the assumption that manipulators have complete information about the non-manipulators. Subsequently, we remove this assumption in Sect. 4.3 to study the impact on complexity of manipulation with top-truncated ballots when there is uncertainty about the non-manipulators' votes.

4.1 Constructive manipulation

Here we look at the complexity of constructive manipulation when top-truncated ballots are allowed. We begin by looking at scoring rules and we completely characterize the complexity

of manipulation for all 3-candidate scoring rules when using each of the evaluation schemes defined before (see Sect. 3.1). Note that in the complete votes case, all scoring rules except plurality are known to be $\mathcal{NP}$-complete for $m \geq 3$ candidates [13,38]. While at initial glance one might think that top-truncated voting may be of limited interest when there are only three candidates, we emphasize that most of our results in this paper are hardness results and thus state that *even* with three candidates certain problems are hard. We argue that this is interesting in itself and provide a foundation for more general results as needed.[1]

4.1.1 Scoring rules

We start with a simple result which shows how computing if a coalition of manipulators can manipulate $X_\uparrow$ (for any scoring rule X), plurality$_\downarrow$, plurality$_{av}$, and veto$_\downarrow$ takes only polynomial time.

Theorem 1 *Computing if a coalition of manipulators can manipulate $X_\uparrow$, veto$_\downarrow$, plurality$_\downarrow$, or plurality$_{av}$, where X is any positional scoring rule, with weighted top-truncated votes takes polynomial time (for any number of candidates).*

Proof For $X_\uparrow$, veto$_\downarrow$, and plurality$_{av}$, the manipulators can simply check if all of them voting for p alone will make it a winner. If not, they cannot make p a winner.

In case of plurality$_\downarrow$, they can check if all of them placing p at the top and all the other candidates in arbitrary order can make p a winner.   □

Next, we look at the complexity of manipulation for any scoring rule that is not isomorphic to plurality or veto (we say that a scoring rule is isomorphic to another if the scoring vector of one is a linear transformation of the other) and when using the round-down evaluation scheme.

Theorem 2 *For any 3-candidate positional scoring protocol X that is not isomorphic to plurality or veto, CWCM with top-truncated votes in $X_\downarrow$ is $\mathcal{NP}$-complete.*

Proof Since there are only three candidates, the scoring vector for the corresponding positional scoring rule is defined by $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$, where $\alpha_1 > \alpha_2 > \alpha_3 = 0$ (because $\alpha_1 = \alpha_2$ is isomorphic to veto, $\alpha_2 = \alpha_3$ is isomorphic to plurality, and α_3 can be taken to be zero since translating the scores in a scoring rule does not affect the outcome of the rule). Also, note that if the three candidates are p, a, and b, each manipulator votes in one of the following ways: $(p > a \sim b)$, $(p > a > b)$, $(p > b > a)$, where for $(p > a \sim b)$ candidate p gets a score α_2.

The problem is in $\mathcal{NP}$ since winner determination for any scoring rule can be done in polynomial time. To show $\mathcal{NP}$-hardness, we proceed by considering three cases: 1) $\alpha_1 > \frac{3}{2}\alpha_2$ 2) $\alpha_1 < \frac{3}{2}\alpha_2$ 3) $\alpha_1 = \frac{3}{2}\alpha_2$. For the first two cases, we show a reduction from the Partition problem, and for the third case we show a reduction from the Fixed-Difference Subset Sum problem.

Case 1 $\alpha_1 > \frac{3}{2}\alpha_2$. Given a Partition instance $\{k_i\}_{1 \leq i \leq t}$ summing to $2K$, construct the following instance of CWCM, where p, a, and b are the three candidates. In S let there be two voters, each of weight $(2\alpha_1 - \alpha_2)K$, voting $(a > b > p)$ and $(b > a > p)$, respectively. As

---

[1] NP-completeness for three candidates does not automatically imply that it is equally hard for a larger number of candidates (for example, see [29, Theorem 4.2] where it is shown that manipulating the 3-veto voting rule is NP-complete for five candidates, but polynomial when there are at least six candidates). Therefore, future work will have to resolve this issue.

a result, a and b have a score of $(2\alpha_1 - \alpha_2)(\alpha_1 + \alpha_2)K$ each. In T let each k_i have a vote of weight $(\alpha_1 + \alpha_2)k_i$.

Suppose there exists a partition. Let those manipulators in one partition vote $(p > a > b)$ and those in the other vote $(p > b > a)$. Then the score of p, a and b, is $2\alpha_1(\alpha_1 + \alpha_2)K$, and so p is a winner (since all of them have the same score and we are considering the non-unique winner model).

Conversely, suppose there exists a manipulation in favour of p. Let x, y, and z be the sum of the k_i's of the manipulators in T who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively. So now, the score of p is $((x + y)\alpha_1 + z\alpha_2)(\alpha_1 + \alpha_2)$, while that of a and b is $((2\alpha_1 - \alpha_2)K + x\alpha_2)(\alpha_1 + \alpha_2)$ and $((2\alpha_1 - \alpha_2)K + y\alpha_2)(\alpha_1 + \alpha_2)$, respectively. Since there exists a successful manipulation, score of p should be at least as large as that of a, and so we have $((x + y)\alpha_1 + z\alpha_2)(\alpha_1 + \alpha_2) \geq ((2\alpha_1 - \alpha_2)K + x\alpha_2)(\alpha_1 + \alpha_2)$. Using the fact that $x + y + z = 2K$, this simplifies to

$$(K - x)\alpha_2 \geq z(\alpha_1 - \alpha_2). \tag{1}$$

Again, the score of p should be at least as large as that of b, so we have

$$(K - y)\alpha_2 \geq z(\alpha_1 - \alpha_2). \tag{2}$$

Adding (1) and (2) and simplifying it, we have $z(2\alpha_1 - 3\alpha_2) \leq 0$. Now, since we assumed $\alpha_1 > \frac{3}{2}\alpha_2$, this implies that $z \leq 0$. But z cannot be less than 0, so it has to be equal to 0. Plugging $z = 0$ in (1) and (2), we have $x \leq K$ and $y \leq K$ respectively. This together with the fact that $x + y + z = 2K$ implies that $x = y = K$, and therefore there exists a partition. *Case 2* $\alpha_1 < \frac{3}{2}\alpha_2$. Given a Partition instance, construct the following instance of CWCM. In S let there be a voter of weight $15K$ voting $(b > a > p)$, a voter of weight $5K$ voting $(b > p > a)$, a voter of weight $11K$ voting $(a > p > b)$, a voter of weight $9K$ voting $(a > b > p)$, a voter of weight $7K$ voting $(p > b > a)$, and a voter of weight $7K$ voting $(p > a > b)$. As a result, the scores of a, b, and p are $(20\alpha_1 + 22\alpha_2)K$, $(20\alpha_1 + 16\alpha_2)K$, and $(14\alpha_1 + 16\alpha_2)K$ respectively. In T let each k_i have a vote of weight $6k_i$.

Suppose there exists a partition. Let those manipulators in one partition (who weight to $6K$) vote $(p > b > a)$ and those in the other vote $(p > a \sim b)$. Then the score of all the three candidates is $20\alpha_1 K + 22\alpha_2 K$, and so p is a winner.

Conversely, suppose there exists a manipulation in favour of p. Let x, y, and z be the total weight of manipulators in T who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$ respectively. So now, the score of p is $(x + y)\alpha_1 + z\alpha_2 + (14\alpha_1 + 16\alpha_2)K$, while that of a and b is $(20\alpha_1 + 22\alpha_2)K + x\alpha_2$ and $(20\alpha_1 + 16\alpha_2)K + y\alpha_2$ respectively. Since there exists a successful manipulation, the score of p should be at least as large as that of a, and so we have $(x + y)\alpha_1 + z\alpha_2 + 14\alpha_1 K + 16\alpha_2 K \geq (20\alpha_1 + 22\alpha_2)K + x\alpha_2$. Using the fact that $x + y + z = 12K$, this simplifies to

$$6(\alpha_1 - \alpha_2)K - x\alpha_2 \geq z(\alpha_1 - \alpha_2). \tag{3}$$

Again, the score of p should be at least as large as that of b, so we have

$$6\alpha_1 K - y\alpha_2 \geq z(\alpha_1 - \alpha_2). \tag{4}$$

Adding (3) and (4) and simplifying it, we have $(6K - z)(2\alpha_1 - 3\alpha_2) \geq 0$. Now, since we assumed $\alpha_1 < \frac{3}{2}\alpha_2$, this implies that $(6K - z) \leq 0$, or $z \geq 6K$. Plugging $z \geq 6K$ in (3) and (4), we have $x \leq 0$, and $y \leq 6K$, respectively. But then x cannot be less than 0, so it has to be equal to 0, and this in turn results in $z \leq 6K$ in (3). This combined with the fact that we already showed z to be greater than or equal to $6K$ implies that z has to be equal to $6K$.

So plugging this in $x + y + z = 12K$, we have $y = 6K$, and this in turn implies there exists a partition.

Case 3 $\alpha_1 = \frac{3}{2}\alpha_2$. Consider the same instance of CWCM as in case 2. The scores of a, b, and p are $(20\alpha_1 + 22\alpha_2)K$, $(20\alpha_1 + 16\alpha_2)K$, and $(14\alpha_1 + 16\alpha_2)K$ respectively. In T let each k_i have a vote of weight $6k_i$.

Suppose there exists S_1, S_2 such that $\sum S_1 - \sum S_2 = K$. Let those manipulators who are in S_1 vote $(p > b > a)$, those in S_2 vote $(p > a > b)$, and let all the remaining manipulators vote $(p > a \sim b)$. If x, y, and z denote the sum of k_i's of the manipulators who vote for $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then the scores of p, a, and b are $9(x + y)\alpha_2 + 6z\alpha_2 + 37\alpha_2 K$, $52\alpha_2 K + 6x\alpha_2$, and $46\alpha_2 K + 6y\alpha_2$, respectively. Now if there existed a manipulation, then the score of p has to be at least as large as that of a and b. Let us consider p and a first. Whatever follows can be replicated for b. Suppose $score(p) \geq score(a)$. This implies $9(x + y)\alpha_2 + 6z\alpha_2 + 37\alpha_2 K \geq 52\alpha_2 K + 6x\alpha_2$. Simplifying this we have, $x + 3y + 2z \geq 5K$. But since $y - x = K$ and $x + y + z = 2K$, we know that $x + 3y + 2z = 5K$, and hence our assumption that $score(p) \geq score(a)$ is true. Doing the same with respect to p and b, we will see that $score(p) = score(b)$. As a result, we can conclude that existence of S_1, S_2 such that $\sum S_1 - \sum S_2 = K$ results in a successful manipulation for p.

Conversely, suppose there exists a manipulation in favour of p. This implies that the score of p is at least as much as that of a, and from above we know that this in turn results in the inequality $x + 3y + 2z \geq 5K$. Using the fact that $x + y + z = 2K$, we have $y - x \geq K$. Similarly, comparing p and b we have, $y - x \leq K$. But then, $y - x$ cannot be both greater and lesser than equal to K at the same time. So $y - x$ has to be equal to K, and this in turn implies that there exists two sets S_1, S_2 such that $\sum S_1 - \sum S_2 = K$, where $y = \sum S_1$ and $x = \sum S_2$. □

Our final result for scoring rules is when the evaluation scheme used is average score.

Theorem 3 *For any 3-candidate positional scoring protocol X that is not isomorphic to plurality, CWCM with top-truncated votes in X_{av} is $\mathcal{NP}$-complete.*

Proof Like in Theorem 2, since there are only three candidates, the scoring vector for the corresponding positional scoring rule is defined by $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$, where $\alpha_1 \geq \alpha_2 > \alpha_3 = 0$. Also, note that if the three candidates are p, a, and b, each manipulator votes in one of the following ways: $(p > a \sim b)$, $(p > a > b)$, $(p > b > a)$, where for $(p > a \sim b)$ candidate p gets a score α_1, a and b receive a score of $(\alpha_2/2)$.

The proof uses a reduction from the Fixed-Difference Subset Sum problem and is very similar to the one in case 3 of Theorem 2. Given an instance of Fixed-Difference Subset Sum, construct the following instance of CWCM where in S we have a voter of weight $(4\alpha_1 + \alpha_2)K$ voting $(b > a > p)$, a voter of weight $(2\alpha_1 - \alpha_2)K$ voting $(a > b > p)$, and a voter of weight $2(\alpha_1 + \alpha_2)K$ voting $(a > p > b)$. As a result, the scores of a, b, and p are $(4\alpha_1 + \alpha_2)(\alpha_1 + \alpha_2)K$, $(4\alpha_1 - \alpha_2)(\alpha_1 + \alpha_2)K$, and $2\alpha_2(\alpha_1 + \alpha_2)K$, respectively. In T let each k_i have a vote of weight $2(\alpha_1 + \alpha_2)k_i$.

Suppose there exists S_1, S_2 such that $\sum S_1 - \sum S_2 = K$. Let those manipulators who are in S_1 vote $(p > b > a)$, those in S_2 vote $(p > a > b)$, and let all those remaining vote $(p > a \sim b)$. If x, y, and z denote the sum of the k_i's of the manipulators who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then the scores of p, a, and b are $(4\alpha_1 + 2\alpha_2)(\alpha_1 + \alpha_2)K$, $((4\alpha_1 + \alpha_2)K + 2(x\alpha_2 + z\alpha_2/2))(\alpha_1 + \alpha_2)$, and $((4\alpha_1 - \alpha_2)K + 2(y\alpha_2 + z\alpha_2/2))(\alpha_1 + \alpha_2)$, respectively. Now if there existed a manipulation, then the score of p has to be at least as large as that of a and b. Let us consider p and a

first. Whatever follows can be replicated for b. Suppose $score(p) \geq score(a)$. This implies that $(4\alpha_1 + 2\alpha_2)K \geq (4\alpha_1 + \alpha_2)K + 2x\alpha_2 + z\alpha_2$. Simplifying this we have, $2x + z \leq K$. But since $y - x = K$ and $x + y + z = 2K$, we know that $2x + z = K$, and hence our assumption that $score(p) \geq score(a)$ is true. Doing the same with respect to p and b, we will see that $score(p) = score(b)$. As a result, we can conclude that existence of S_1, S_2 such that $\sum S_1 - \sum S_2 = K$ results in a successful manipulation for p.

Conversely, suppose there exists a manipulation in favour of p. This implies that the score of p is at least as much as that of a, and from above we know that this in turn results in the inequality

$$2x + z \leq K. \tag{5}$$

Similarly, comparing p and b we have,

$$2y + z \leq 3K. \tag{6}$$

Now, using the fact that $x + y + z = 2K$, we know that inequality (5) reduces to $y - x \geq K$ and inequality (6) reduces to $y - x \leq K$. But then, $y - x$ cannot be both greater and lesser than equal to K at the same time. So $y - x$ has to be equal to K, and this in turn implies that there exists two sets S_1, S_2 such that $\sum S_1 - \sum S_2 = K$, where $y = \sum S_1$ and $x = \sum S_2$. □

4.1.2 Scoring elimination rules

We now consider scoring elimination rules and first look at how top-truncated voting affects the complexity of manipulation in eliminate(Veto). Following that we prove a general result for all scoring elimination rules.

Theorem 4 *For eliminate(Veto), in the unique winner model, any manipulation that can be achieved by casting top-truncated votes can be achieved if only complete votes were allowed.*

Proof Consider an arbitrary set W of top-truncated votes which—along with the set S of non-manipulators' votes—results in an elimination order $e = (c_1, c_2, \ldots, c_m = p)$, where p is the preferred candidate, and c_i is the candidate eliminated in the ith round. Now, consider the set of votes X such that each vote in W is replaced by $(p = c_m > c_{m-1} > \cdots > c_1)$. X along with S results in the same elimination order e. Therefore, we see that any manipulation that can be achieved by a set of top-truncated votes can be achieved by casting an equivalent set of complete votes. □

Since top-truncated voting does not encourage more strategic voting, it follows that for bounded number of candidates—i.e., when there are only a (small) fixed number of candidates—we can use the result by Coleman and Teague who showed that CWCM for eliminate(Veto) is in $\mathcal{P}$ when the votes are complete [12, Theorem 11].

Corollary 5 *In the unique winner model, computing if a coalition of manipulators can manipulate eliminate(Veto) with weighted top-truncated votes takes polynomial time for bounded number of candidates.*

Next, we consider elimination versions of 3-candidate scoring rules in general and show that CWCM with top-truncated votes is $\mathcal{NP}$-complete for elimination version of any scoring rule that is not isomorphic to veto. For this, we first show that top-truncated voting does not change the complexity of a related problem, Anti-WCM, for any scoring rule.

Definition 7 (*Anti-WCM* [12]) Given a set S of weighted votes, the weights for a set of votes T, and a disliked candidate d, we are asked if there exists a way to cast the votes in T so that it results in d receiving the lowest score.

Theorem 6 *When using the round-up evaluation scheme, top-truncated voting does not change the worst-case complexity of Anti-WCM for any scoring protocol.*

Proof Assume there exists an arbitrary set W of top-truncated votes which results in d receiving the lowest score. For each of the top-truncated votes in W, let us complete them in the following way: If d is included in the vote, append all the other candidates who are not part of it in any arbitrary order. If d is not there, place d at the bottom of the preference ordering (as the mth preferred candidate) and the rest in any arbitrary order. Completing the votes as above does not change the candidate with the lowest score.   □

As a result of the above theorem, we have the following corollary which says that for any scoring rule not isomorphic to veto, Anti-WCM with top-truncated votes is $\mathcal{NP}$-complete. Note that the corollary here is based on the result of Coleman and Teague who proved that Anti-WCM is $\mathcal{NP}$-complete for all scoring rules not isomorphic to veto [12, Corollary 10.1].

Corollary 7 *For any scoring rule with $\alpha = \langle \alpha_1, \ldots, \alpha_m \rangle$, Anti-WCM with top-truncated votes is in $\mathcal{P}$ if $\alpha_1 = \cdots = \alpha_{m-1}$ and is $\mathcal{NP}$-complete otherwise.*

Next, we use the above result to show our main result concerning all scoring elimination rules except eliminate(Veto).

Theorem 8 *For any 3-candidate scoring rule X that is not isomorphic to veto, CWCM with top-truncated votes in eliminate(X) is $\mathcal{NP}$-complete.*

Proof Since there are only three candidates and the corresponding positional scoring rule is not isomorphic to veto, the scoring vector for the same is defined by $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$, where $\alpha_1 > \alpha_2 \geq \alpha_3 = 0$. Showing that the problem is in $\mathcal{NP}$ is easy. To show $\mathcal{NP}$-hardness, we use a reduction from the Anti-WCM problem. Given an arbitrary instance of Anti-WCM$\langle S, T, h \rangle$ with 3 candidates, where a, b, and h are the three candidates, and h is the disliked candidate, we construct an instance of CWCM with the same set of candidates, the same set of manipulators T, and to the S from the Anti-WCM instance we add the following set S' of voters such that K is greater than the sum of the weights in S and T combined. In each case, we add 1 voter of the corresponding type and weight specified below.

$$K : (a > b > h)$$
$$K : (b > h > a)$$
$$K : (h > a > b)$$

We set a to be the preferred candidate in the CWCM instance.

Suppose there was a way to make h receive the lowest score in the Anti-WCM instance. This implies that h receives the lowest score in the CWCM instance since all the three candidates a, b, and h are tied in S' (each of them receives a score of $\alpha_1 K + \alpha_2 K$ from S'). Therefore, h will be eliminated in the first round, and following that the votes in S' would be:

$$K : (a > b)$$
$$K : (b > a)$$
$$K : (a > b)$$

And now since there are only two candidates, for any scoring rule X, eliminate(X) is equivalent to eliminate(Veto). Hence, the candidate with the most number of last preferences will be eliminated next. In our case this is b since it has K extra last preferences. Therefore, a wins, thus ensuring a successful manipulation in the CWCM instance.

Conversely, suppose there exists a successful constructive manipulation for a in the CWCM instance. Now, this is possible only if h is eliminated in the first round, because otherwise if b is eliminated in the first round, then a will have K extra last preferences thus resulting in its elimination in the second round. This in turn implies that there is a successful manipulation against h in the Anti-WCM instance.   □

Since the plurality with runoff rule is the same as STV when there are only three candidates, we have the following corollary which also follows from the result of Narodytska and Walsh [42, Proposition 6].

Corollary 9 *For the 3-candidate plurality with runoff rule, CWCM with top-truncated votes is $\mathcal{NP}$-complete.*

4.1.3 Copeland$^\alpha$

Narodytska and Walsh [42] showed that CWCM with top-truncated votes in the Copeland rule (Copeland$^{0.5}$) is $\mathcal{NP}$-complete for four candidates. Additionally, they also conjectured that the result holds when the number of candidates is three. Here we prove that conjecture, and also show that our hardness result holds for all rational $\alpha \in [0, 1)$. We note that the following result was also independently obtained by Fitzsimmons and Hemaspaandra [33].

Theorem 10 *Let α be a rational number with $0 \le \alpha < 1$. For Copeland$^\alpha$, CWCM with top-truncated votes is $\mathcal{NP}$-complete for three candidates.*

Proof It is easy to show that the problem is in $\mathcal{NP}$. To show that it is $\mathcal{NP}$-hard, we use a reduction from the Fixed-Difference Subset Sum problem. Given an arbitrary instance of Fixed-Difference Subset Sum, construct an instance of CWCM where p, a, and b are the three candidates. In S let there be a voter of weight $3K$ voting $(a > b > p)$ and a voter of weight K voting $(b > a > p)$. In T let each k_i have a vote of weight $2k_i$.

Suppose there exists S_1, S_2 such that $\sum S_1 - \sum S_2 = K$. In Copeland$^\alpha$, it can be assumed that all the manipulators rank p first. So, let the manipulators in S_1 vote $(p > b > a)$, those in S_2 vote $(p > a > b)$, and let the rest vote for $(p > a \sim b)$. If $N_V(r, s)$ denotes the total number of votes in V which rank r prior to s and $D_V(r, s) = N_V(r, s) - N_V(s, r)$, then $D_{S \cup T}(p, a) = 0$ and $D_{S \cup T}(p, b) = 0$. Therefore, the score of p is 2α. Also since $\sum S_1 - \sum S_2 = K$, $D_T(a, b) = -2K$, while $D_S(a, b) = 2K$. Therefore, $D_{S \cup T}(a, b) = 0$ and so, both receive a score 2α. Since all of them have the same score, p is a winner.

Conversely, suppose there exists a successful manipulation in favour of p. If x, y, and z, denote the sum of k_i's of the manipulators in T who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then without taking into account the pairwise election between a and b in T the score of p, a, and b is $2\alpha, 1 + \alpha$, and α, respectively. Now since $2\alpha < 1 + \alpha$ for all rational $\alpha \in [0, 1)$, therefore, the only way p would win this is if including the pairwise election between a and b in T results in a tie between them. So this implies that $D_{S \cup T}(a, b) = 2K + 2x - 2y = 0$ and that $y - x = K$. This in turn implies that there exists sets S_1 and S_2 such that $\sum S_1 - \sum S_2 = K$, where $y = \sum S_1$ and $x = \sum S_2$.   □

4.1.4 Maximin

Although we have seen instances like in scoring rules with the round-up evaluation scheme (Theorem 1) where top-truncated voting decreases the complexity of manipulation (as compared to the complete votes case), they aren't really conclusive in the sense that one could question the choice of the evaluation scheme that in turn caused the result. Clearly, the round-up evaluation scheme isn't a good choice as what we're essentially doing by employing the same is to encourage the manipulators to treat it just as a plurality-type rule. Therefore, one question that could arise is: "Is there a voting system for which no matter how the top-truncated votes are dealt with the complexity of manipulation with top-truncated ballots decreases?". We answer this question below in the affirmative. We see that as long as all the unranked candidates are considered tied and are assumed to be ranked below the ranked candidates (which is the natural definition of a top-truncated vote), the complexity of manipulation with top-truncated ballots is in $\mathcal{P}$ for the maximin rule. Note that CWCM for the maximin rule is known to be $\mathcal{NP}$-complete for four candidates when we consider only complete votes [13, Theorem 8].[2]

Theorem 11 *Computing if a coalition of manipulators can manipulate the maximin protocol with weighted top-truncated votes takes polynomial time (for any number of candidates).*

Proof Let W be an arbitrary set of top-truncated votes which—along with the set S of non-manipulators' votes—results in p being a winner. Since in the maximin rule moving p to the top will never hurt p, we can safely assume that all the votes in W have p at the top. Now for every vote in W, replace it by $(p > c_1 \sim \cdots \sim c_{m-1})$, where $c_1, \ldots, c_{m-1}$ are the other candidates in an m-candidate election. By doing so we see that p's score does not change. Also, note that for all the other candidates their scores can only decrease or stay the same, but can never increase. Therefore, any constructive manipulation achieved for p can be achieved if all manipulators just vote $(p > c_1 \sim \cdots \sim c_{m-1})$. □

4.2 Destructive manipulation

In this section, we look at the complexity of destructive manipulation when top-truncated ballots are allowed. We begin by looking at a broad class of rules for which top-truncated voting has no impact on strategic voting. This class consists of all voting rules that are i) based on numerical scores—meaning that the candidates are assigned numerical scores based on the votes, ii) where the candidates with the highest score form the winner set, and iii) are score-monotone, meaning that if a voter v_i changes his or her vote (from $>$ to $>'$) in such a way that $\{b : a > b\} \subseteq \{b : a >' b\}$, then a's score will not decrease (or informally, more support for a candidate will not decrease it's score).

Theorem 12 *For any voting rule that is based on numerical scores, where the candidates with the highest score forms the winner set, and is score-monotone, any destructive manipulation that can be achieved by casting top-truncated votes can be achieved if only complete votes were allowed.*

Proof Consider such a voting rule X. Let the destructive manipulation be against the candidate h. Now, suppose there exists an arbitrary set of top-truncated votes W that—along with the set S of non-manipulators' votes—results in the destructive manipulation of h in X.

---

[2] Although Conitzer et al. [13] use the unique winner model, it can be verified that the result holds for the non-unique winner model as well.

Since X is based on scores, we can list the candidates in the decreasing order of their scores. Let $c_1(\neq h), c_2, \ldots, c_m$ be this ordering. Next, consider the set of votes W' which is formed by completing the votes in W in the following way: replace each vote in W by placing c_1 at the top, h at the bottom (i.e., at the mth position), and the rest of the candidates in any arbitrary order. Since X is score-monotone, W' along with S cannot result in the score of c_1 decreasing and nor can it result in the score of h increasing. Therefore, if W resulted in the destructive manipulation of h, then so should W'.   □

Since top-truncated voting has no impact on destructive manipulation in rules that are based on numerical scores, where the candidates with the highest score forms the winner set, and are score-monotone, it follows that for bounded number of candidates we can use the result by Conitzer et al. [13] who showed that DWCM was in $\mathcal{P}$ for all such rules when only complete votes are allowed.

Corollary 13 *DWCM with top-truncated votes is in $\mathcal{P}$ for all scoring rules, for the maximin rule, and for Copeland$^\alpha$.*

Next, we consider the elimination versions of all 3-candidate scoring rules and we show how for all scoring rules that are not isomorphic to veto DWCM is $\mathcal{NP}$-complete. We prove this by using a similar reduction as in Theorem 8 from an arbitrary instance of Anti-WCM.

Theorem 14 *For any 3-candidate positional scoring rule X that is not isomorphic to veto, DWCM in eliminate(X) with top-truncated votes is $\mathcal{NP}$-complete.*

Proof Since there are only three candidates, the scoring vector for the corresponding positional scoring rule is defined by $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$, where $\alpha_1 > \alpha_2 \geq \alpha_3 = 0$. Showing that the problem is in $\mathcal{NP}$ is easy. To show $\mathcal{NP}$-hardness, we use a reduction from the Anti-WCM problem (see Corollary 7). Given an arbitrary instance of Anti-WCM$\langle S, T, h \rangle$ with 3 candidates, where a, b, and h are the three candidates, and h is the disliked candidate, we construct an instance of DWCM with the same set of candidates, the same set of manipulators T, and to the S from the Anti-WCM instance we add the following set S' of voters such that K is greater than the sum of the weights in S and T combined. In each case, we add 1 voter of the corresponding type and weight specified below.

$$K : (a > h > b)$$
$$2K : (h > a > b)$$
$$K : (b > h > a)$$
$$2K : (h > b > a)$$
$$3K : (a > b \sim h)$$
$$3K : (b > a \sim h)$$

We set h to be the disliked candidate in the DWCM instance.

Suppose there was a way to make h receive the lowest score in the Anti-WCM instance. If score$_S(a)$ denotes the score candidate a receives from S, then this implies that score$_{S \cup T}(a)$ > score$_{S \cup T}(h)$, and score$_{S \cup T}(b)$ > score$_{S \cup T}(h)$. Also, note that all the three candidates a, b, and h are tied in S' as each of them receive a score of $4\alpha_1 K + 2\alpha_2 K$. This in turn implies that h receives the lowest score in the DWCM instance, and so will be eliminated in the first round. Thus, existence of a successful manipulation in the Anti-WCM instance ensures the existence of a successful manipulation in the DWCM instance.

Conversely, suppose there exists a successful destructive manipulation against h in the DWCM instance. We first show that this is possible only if h is eliminated in the first round in eliminate(X). To do so, let us assume it were not the case and that one of a or b was eliminated in the first round. Let us consider a first. If a was eliminated in the first round then the votes in S' would now be:

$$K : (h > b)$$
$$2K : (h > b)$$
$$K : (b > h)$$
$$2K : (h > b)$$
$$3K : (b > h)$$

Since the elimination of a means that there are only two candidates remaining, from now on we can assume our protocol to be equivalent to plurality (since any scoring rule is equivalent to plurality when there are only two candidates). So now, $\text{score}_{S'}(h) - \text{score}_{S'}(b) = K$ and since K is greater than sum of the weights in S and T combined this implies that in the subsequent round b will be eliminated, thus resulting in h winning the DWCM instance. Therefore, there cannot be a destructive manipulation against h in the DWCM instance if a is eliminated in the first round. Similarly, by doing things identically for candidate b, we can see that a destructive manipulation against h will not be possible if b is eliminated in the first round. This in turn leads us to conclude that a successful destructive manipulation against h is possible only if h is eliminated in the first round. But then, since all the three candidates are tied in S', the only way this can happen is if h receives the lowest score in S. Or in other words, a successful destructive manipulation against h in the DWCM instance is possible only if there exists a successful manipulation against h in the Anti-WCM instance. □

Again, since the plurality with runoff rule is equivalent to STV when there are only three candidates, we have the following corollary.

Corollary 15 *For the 3-candidate plurality with runoff rule, DWCM with top-truncated votes is $\mathcal{NP}$-complete.*

4.3 Impact on complexity when there is uncertainty about the non-manipulators' votes

So far, we have looked at the complexity of manipulation with top-truncated ballots when the manipulators have complete information on the non-manipulators' votes. Although this is a useful setting to study, especially given that it enables us to look at the hardness of manipulation without having to worry about the complexities that are introduced as part of the uncertainty model, the assumption that the manipulators will have complete information isn't always realistic. Therefore, we now look at how incomplete information about the non-manipulators impacts the complexity of manipulation with top-truncated ballots.

To model the incomplete information setting, we consider the following two scenarios and we study each of them separately.

1. What if only top-truncated preference orderings of the non-manipulators were visible to the manipulators?
2. What if the manipulators have only probabilistic information on the votes of the non-manipulators?

Before we go on to the results, we note that the partial information assumption has been considered before. For instance, Conitzer et al. [13] considered the second setting for the case of complete votes, and Briskorn et al. [10] and Erdélyi and Reger [22] looked at the problem of bribery under different models of partial information. Additionally, there is also recent work by Dey et al. [16] which looks at the complexity of manipulation under the partial information assumption. Here they define three notions of manipulation and of these we note that the Weak Manipulation (WM) notion that they define is a generalization to the first scenario that we consider here. However, we also note that there are no overlaps between the results here and in that paper since their results on the WM problem are on voting rules that are not considered here.

4.3.1 When only top orders of the non-manipulators are visible

Before we look at the problem of manipulation, let us introduce the following problem which is the weighted version of the extension-bribery problem which in turn was introduced by Baumeister et al. [6]. Informally, the Weighted Extension-bribery problem captures one of the tasks a campaign manager faces in the presence of truncated ballots and it is concerned with trying to figure out if it possible to extend a partially specified vote—as in, if it is possible to increase the number of ranked candidates specified in the preference order by suitably filling it up—so as to ensure that the manager's candidate wins. Now, since it is natural that not all voters can be easily convinced, this entails that we assume that every voter i has an associated cost function. This cost function for every agent i, which is assumed to be dependent only on the number of added candidates, is the δ_i in the definition below. More formally, the assumption is that $\delta_i : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}^+ \cup \{0\}, \delta_i(0) = 0$, and δ_i is non-decreasing (at this point we refer the reader to Baumeister et al. [6] for more a detailed discussion on the extension-bribery problem).

Definition 8 (*Weighted extension-bribery*) We are given a set S of votes which are possibly top-truncated, the weights of each of the voters in V, a collection $\Delta = (\delta_1, \ldots, \delta_n)$ of extension-bribery cost functions, a preferred candidate p, and a non-negative integer B (which is the budget). We are asked if there exists an extension to the votes in S with cost $\leq B$ such that p is the winner.

Here we are concerned with only a special case of the Weighted Extension-bribery which we refer to as the Weighted Extension-bribery with zero costs where $\delta_i = 0, \ 1 \leq i \leq n$.

Next, before we look at the complexity of Weighted Extension-bribery with zero costs for all the voting rules considered in this section, consider the following version (CWCM*) of CWCM with top-truncated votes where the only difference is that here we make it necessary for the non-manipulators to always have complete ballots (i.e., in CWCM* only the manipulators can cast top-truncated votes).

Definition 9 (*CWCM**) CWCM* with top-truncated votes is the same problem as CWCM with top-truncated votes, with the additional restriction that the non-manipulators always have complete preference orders.

The first thing to observe here is that all the $\mathcal{NP}$-complete results for CWCM with top-truncated votes from Sect. 4.1 hold for CWCM* as well, since a close look at the proofs for Theorems 2, 3, 8, and Theorem 10 reveal that in all the cases we showed reductions to instances which always had complete orders for the non-manipulators. Hence, we have the following theorem.

Theorem 16 *CWCM* with top-truncated votes is $\mathcal{NP}$-complete for 3-candidate $X^1_\downarrow$, 3-candidate X^2_{av}, eliminate(X^3), 3-candidate Copeland$^\alpha$ for $\alpha \in [0, 1)$, and 3-candidate plurality with runoff rule, where X^1 represents all scoring rules except plurality and veto, X^2 represents all scoring rules except plurality, and X^3 represents all scoring rules except veto.*

The second observation to make regarding CWCM* with top-truncated votes is that it is a special case of Weighted Extension-bribery with zero costs, where some of the ballots are complete (those of the non-manipulators) and some are empty (those of the manipulators). Therefore, any hardness result for CWCM* carries over for Weighted Extension-bribery with zero costs, and so we have the following theorem.

Theorem 17 *Weighted extension-bribery problem with zero costs is $\mathcal{NP}$-complete for 3-candidate $X^1_\downarrow$, 3-candidate X^2_{av}, eliminate(X^3), 3-candidate plurality with runoff, and 3-candidate Copeland$^\alpha$ for $\alpha \in [0, 1)$, where X^1 represents all scoring rules except plurality and veto, X^2 represents all scoring rules except plurality, and X^3 represents all scoring rules except veto.*

Additionally, we also have the following result which shows that Weighted Extension-bribery with zero costs is in $\mathcal{P}$ for certain voting rules considered here.

Theorem 18 *Weighted Extension-bribery with zero costs is in $\mathcal{P}$ for $X_\uparrow$, $veto_\downarrow$, $plurality_\downarrow$, $plurality_{av}$, and the maximin protocol, where X is any positional scoring rule.*

Proof For all the above mentioned protocols except plurality$_\downarrow$, the best we can do is to extend each top-truncated vote by placing p at its end (i.e., place p as kth candidate if $k - 1$ candidates are already ranked), if it isn't already present. In case of plurality$_\downarrow$, the best strategy is to complete each of the top-truncated votes by placing p at the topmost position possible followed by all the other as-yet unranked candidates in any arbitrary order. □

Next we show that if Weighted Extension-bribery with zero costs is hard then constructive manipulation with even a single manipulator is hard.

Definition 10 (*CWIM-TTU*) In Constructive Weighted Individual Manipulation under Top-Truncated Uncertainty (CWIM-TTU), we are given a set S of partial votes of the non-manipulators, the weight of the manipulator, and a preferred candidate p. We are asked if the manipulator can cast his vote in such a way so as to ensure that there is at least one extension to the votes in S that makes p a winner.

Theorem 19 *If Weighted Extension-bribery with zero costs is $\mathcal{NP}$-hard for a given protocol with k candidates, then CWIM-TTU with top-truncated votes is also $\mathcal{NP}$-hard for it with k candidates.*

Proof Construct an instance of CWIM-TTU from an Weighted Extension-bribery with zero costs instance by just adding a manipulator of weight 0.

Combining Theorems 17 and 19, we have the following result which says that for all the protocols considered in Sect. 4.1 for which CWCM with top-truncated ballots was hard, even individual manipulation with top-truncated votes is hard when there is uncertainty about the non-manipulators' votes.

Theorem 20 *CWIM-TTU with top-truncated votes is $\mathcal{NP}$-complete for 3-candi-date $X^1_\downarrow$, 3-candidate X^2_{av}, eliminate(X^3), 3-candidate plurality with runoff rule, and 3-candidate Copeland$^\alpha$ for $\alpha \in [0, 1)$, where X^1 represents all scoring rules except plurality and veto, X^2 represents all scoring rules except plurality, and X^3 represents all scoring rules except veto.*

Finally, we conclude this section by showing that the CWIM-TTU with top-truncated votes is in $\mathcal{P}$ for eliminate(Veto) (it is easy to see that it is also in $\mathcal{P}$ for all the rules mentioned in Theorem 18). We show this by first proving that Weighted Extension-bribery with zero costs is in $\mathcal{P}$ for eliminate(Veto).

Theorem 21 *In the unique winner model, Weighted Extension-bribery with zero costs is in $\mathcal{P}$ for eliminate(Veto) when the number of candidates is bounded.*

Proof Suppose there was an arbitrary extension W which resulted in p winning. Let the corresponding elimination order be $e = (c_1, c_2, \ldots, c_m(= p))$, where c_i is the candidate eliminated in the ith round. Now, we claim that the same elimination order can be achieved by doing the following:

– In each of the top-truncated votes, complete it by placing the candidates in the reverse order in which they appear in e. That is, place p if not already present, followed by c_{m-1} if not present, and so on until c_1.

Doing the above results in the same elimination order as e. This can be shown through an inductive argument. When there are m candidates, c_1 which is eliminated first in e has been placed last wherever possible and this in turn will result in it getting eliminated first in our completion. Once c_1 is eliminated, we have $m - 1$ candidates with c_2 placed last wherever possible and therefore c_2 will be eliminated next. Continuing this way, guarantees the elimination order e.

To solve Weighted Extension-bribery, the campaign manager can try out all possible elimination orders, extend the votes as outlined above, and see if any of them results in p winning. Doing so will only take polynomial time since the number of candidates is bounded. □

Theorem 22 *In the unique winner model, CWIM-TTU with top-truncated votes is in $\mathcal{P}$ for eliminate(Veto) when the number of candidates is bounded.*

Proof Since the Weighted Extension-bribery with zero costs problem is in $\mathcal{P}$, the manipulator can try out all possible complete orders to check if manipulation is possible. This is enough because, for eliminate(Veto), any manipulation that can be achieved by top-truncated votes can be achieved by completing the votes appropriately (see Theorem 4). □

The above result also implies that, in the unique winner model, CWCM under Top-Truncated Uncertainty and with top-truncated votes is in $\mathcal{P}$ for eliminate(Veto) when the number of candidates is bounded, since in eliminate(Veto) any manipulation that can be induced by an arbitrary set of the manipulators' vote can be induced if all the manipulators vote in the same way [12, Lemma 12].

4.3.2 When there is only probabilistic information on the non-manipulators

Manipulation under probabilistic uncertainty was introduced by Conitzer et al. [13]. In their work they introduced the Weighted Evaluation problem which, given a probability distribution on the votes, and a number $r \in [0, 1]$, asks if the probability of a candidate winning

is greater than r. Subsequently, they also proved that if CWCM (with complete votes) for a voting protocol is hard then so is Weighted Evaluation [13, Theorem 15]. Now, since we additionally allow top-truncated votes, we can state the following result which is almost equivalent to [13, Theorem 15] with the only difference being that the "set of all possible votes" would now contain top-truncated votes as well and that the reduction here is from CWCM with top-truncated votes. In fact, most of the results in this section are only extensions to the corresponding result from Conitzer et al.'s paper [13] that arise as a result of allowing top-truncated voting.

Theorem 23 *If CWCM with top-truncated votes is $\mathcal{NP}$-hard for a given protocol with k candidates, then Weighted Evaluation when top-truncated votes are allowed is also $\mathcal{NP}$-hard for it (with k candidates) even if $r = 0$, the votes are drawn independently, and only the following types of distributions are allowed: (1) the vote's distribution is uniform over all possible votes, or (2) the vote's distribution puts all of the probability mass on a single vote.*

Proof We can proceed exactly as in [13, Theorem 15] to prove this result. To begin, consider an arbitrary instance of CWCM with top-truncated votes with the set S of non-manipulators, T of manipulators, and p, the preferred candidate. Let the Weighted Evaluation instance be the same, with $r = 0$, and p being the candidate to be evaluated. Now, for the voters in S, since their votes are already given in the CWCM instance, in the Weighted Evaluation instance its distribution places all of the probability mass on that given vote. On the other hand, if the voter was in T, then its vote is drawn uniformly from the set of all the possible votes. Now, it is easy to see that p wins with a probability greater than zero in the Weighted Evaluation instance if and only if there exists some way to cast votes for voters in T such that there is a successful constructive manipulation in favour of p in the CWCM instance. □

Corollary 24 *Weighted Evaluation when top-truncated votes are allowed is $\mathcal{NP}$-hard for 3-candidate $X_{\downarrow}^1$, 3-candidate X_{av}^2, eliminate(X^3), 3-candidate plurality with runoff, and 3-candidate Copeland$^\alpha$ for $\alpha \in [0, 1)$, when the votes are drawn independently, and the distributions allowed are: (1) uniform over all possible votes, or (2) the vote's distribution puts all of the probability mass on a single vote, where X^1 represents all scoring rules except plurality and veto, X^2 represents all scoring rules except plurality, and X^3 represents all scoring rules except veto.*

Next, we show a relation between the Weighted Evaluation and manipulation with a single manipulator as in [13]. Note that the difference between CWIM with Uncertainty (CWIM-U) that we define below and CWIM-TTU that we defined in the Sect. 4.3.1 is in how the partial information about the non-manipulators' votes is specified.

Definition 11 (*CWIM-U*) In Constructive Weighted Individual Manipulation under Uncertainty (CWIM-U), given a distribution over all the non manipulators' votes, the weights of the non-manipulators, the weight of the manipulator, a preferred candidate, p, and a number $r \in [0, 1]$, we are asked if the manipulator can cast his vote in such a way so as to ensure p wins with a probability greater than r.

Theorem 25 *If Weighted Evaluation with top-truncated votes is $\mathcal{NP}$-hard for a protocol with k candidates and some restrictions on the distribution, then CWIM-U with top-truncated votes is also $\mathcal{NP}$-hard for the same protocol with k candidates and the same restrictions on the distribution.*

Proof Construct an instance of CWIM-U from an arbitrary Weighted Evaluation instance by just adding a manipulator of weight 0. □

Finally, we show that Weighted Evaluation when top-truncated votes are allowed can be hard even if CWCM with top-truncated votes is in $\mathcal{P}$. To do so, we consider eliminate(Veto) for which CWCM with top-truncated votes was shown to be in $\mathcal{P}$ for bounded number of candidates (see Corollary 5).

Theorem 26 *Weighted Evaluation when top-truncated votes are allowed is $\mathcal{NP}$-hard in 3-candidate eliminate(Veto) even if $r = 0$, the votes are drawn independently, and the distribution over each vote has a positive probability for at most 2 of the votes.*

Proof We show this by a reduction from the Partition problem. Given an instance of Partition, construct an instance of Weighted Evaluation where a, b, and p are the three candidates. Let there be a vote of weight 1 for $(p > a > b)$. For each k_i in the partition instance, let it have a weight k_i and vote for $(a > p > b)$ and $(b > p > a)$ with probability 1/2 each.

Let x and y denote the sum of the weights of the voters voting $(a > p > b)$ and $(b > p > a)$, respectively. Next, consider the case when $x \geq K + 1$. If this is the case, then since $x + y = 2K$, this implies that $y \leq K - 1$ and so p will get eliminated in the second round. On other hand if $x \leq K - 1$, then $y \geq K + 1$ and so again p will get eliminated in the second round. Hence we see that p wins if and only if $(a > p > b)$ and $(b > p > a)$ are voted by exactly K of the vote weight as failing to do so would mean that p will be eliminated in the second round. But then this is possible if and only if there exists a partition.

□

5 Reinstating combinatorial protections for manipulation and bribery in single-peaked electorates

In the previous section we looked at the complexity of manipulation when the voters are allowed to cast top-truncated ballots. The fundamental assumption in all of the results was that there is no particular structure in the agents' preferences, and so, given the set of candidates, C, it was assumed that an agent can specify any linear order over C or a non-empty subset of it. However, there are many election scenarios where the preferences of agents have an underlying structure, meaning that there is a subset of admissible votes that are never cast. For instance, there have been studies that have shown that many important election scenarios like the U.S. presidential elections and elections in committees are close to having a single-peaked structure (see, for example, [8,43]).

One natural question that arises in the above context is: "What happens to the host of results in computational social choice where $\mathcal{NP}$-hardness shields were obtained for different voting rules using constructions on combinatorially rich structures such as partitions and covers? Do they still hold when the electorate has structured preferences?". This question was first raised by Walsh [47] and was subsequently studied in detail by Faliszewski et al. [29] for manipulation and control in single-peaked electorates and by Brandt et al. [9] for bribery in single-peaked electorates. The overarching theme in all these papers was that the previously obtained combinatorial protections vanish when the voters have structured preferences—i.e., many of the $\mathcal{NP}$-hardness results fall into polynomial time.

In light of the above results, our results in this section take this line of research in a new direction by looking at the impact of partial preferences on manipulative actions in single-peaked electorates. In particular, we consider top-truncated preferences, and we look at their impact on manipulative actions in single-peaked settings. In doing so, we arrive at a number of surprising results, which in turn form the theme of this section—of reinstating combinatorial protections by allowing top-truncated voting. Although, as noted by Faliszewski et al.

[30], polynomial time algorithms for manipulative actions such as bribery and control are not always unethical or "bad" (as it can be a valuable tool in the hands of, say, the campaign manager, committee chair etc.), in the majority of situations they aren't socially "good" either and hence we believe that it is important to have elections protected against them. This thought, along with the "easiness" results previously obtained, and the fact that, among structured preference profiles, single-peaked preferences are the most widely studied, forms the main motivation of our work here. The overarching theme in this section is that top-truncated voting is useful in reinstating combinatorial protections in single-peaked electorates, and we believe that these results form a *win-win* scenario: allowing voters to specify top-truncated ballots (or partial preferences, in general) is extremely useful and often necessary in many multi-agent systems applications and even in real-world elections, and allowing this additional flexibility in turn gives us what we want in terms of making the complexity of many manipulative-action problems hard.

The rest of this section is organized as follows. First, in Sect. 5.1, we formally define single-peaked preferences. Subsequently, in Sect. 5.2, we present all our results for manipulation when the electorate is single-peaked. Following this we turn to the problem of weighted bribery in Sect. 5.3. Finally, in Sect. 5.4, we address the question on whether our observations—of top-truncated voting reinstating combinatorial protections—are universal and ask if there are voting rules for which top-truncated voting is not beneficial in terms of increasing the complexity of manipulation.

5.1 Single-peaked preferences

Most work in computational social choice assumes no particular structure for an agent's preference—i.e., given the set of candidates C, it is assumed that an agent can specify any linear order over C or a non-empty subset of it (in the case of truncated orders). However, there are many election scenarios where the preferences of agents have an underlying structure and therefore there are some admissible votes that would never be cast. So, in this context, we consider one of the most widely studied such notions (see, e.g., [9,29])—single-peaked preferences.

The notion of single-peaked preferences, first introduced by Black [7], captures settings where the preferences of a voter are based on a one-dimensional axis. The basic idea here is that every voter has a peak (their most preferred alternative) and that their utility for an alternative decreases the further it is away from this peak. More formally, a preference profile, P, where the preference, $\succ_v$, of each $v \in V$ is a linear order over C, is said to be single-peaked if there exists a linear order L over C such that for every triple of candidates a, b, and c it holds that

$$(a \; L \; b \; L \; c \lor c \; L \; b \; L \; a) \implies (\forall v \in V) \, [a >_v b \implies b >_v c] .$$

Although the above definition of single-peakedness assumes that the voters have complete orders, when voters are allowed to present top-truncated ballots, this notion of single-peakedness essentially captures those scenarios where they have a continuous range over L over which their preferences are single-peaked and outside of which they are indifferent among the alternatives. In social choice theory, this notion of single-peaked with top-truncated preferences has been captured as "single-peaked with outside options" in the context of choosing a level of public good by Cantala [11].

Example 2 Let $C = \{c_1, c_2, c_3, c_4\}$ and $c_2 \; L \; c_4 \; L \; c_3 \; L \; c_1$ be a linear order over C. Then, the preference orders $(c_4 > c_3 > c_2 > c_1)$ and $(c_4 > c_2 > c_3 > c_1)$ are both valid complete

single-peaked orders, while the preference order ($c_4 > c_1 > c_2 > c_3$) is not a valid single-peaked order. Also, with respect to the given linear order, ($c_4 > c_3 > c_2 \sim c_1$) and ($c_4 > c_2 > c_3 \sim c_1$) are both valid top-truncated single-peaked orders, while ($c_4 > c_1 > c_2 \sim c_3$) is not.

In addition to the model of Cantala [11], Lackner [40] also introduced another model that extends the definition of single-peaked preferences to accommodate partial orders. In this model a profile P of partial orders is said to be single-peaked with respect to an axis L if each of the preference orders in P can be extended to a total order that is single-peaked with respect to L. We note that when restricted to the case of top-truncated preferences, the model of Lackner [40] is equivalent to the one of Cantala [11] (which is the one we use here).

When discussing results for manipulative action problems in single-peaked electorates, we will follow the model proposed by Walsh [47] and hence assume that the societal order L is given as part of the input—or in other words, we assume that the unidimensional issue of concern to an electorate is publicly known (for a more in-depth discussion on why this is a reasonable, and often useful, assumption, see, for example, [9, Section 2] and [30, Section 4]).

5.2 Manipulation in single-peaked electorates

In this section we study CWCM with top-truncated votes in single-peaked electorates. Since the theme here is the reinstatement of combinatorial protections by top-truncated voting, for all the voting rules considered in this section, we present both the "easiness" result (if not already known from previous work) as well as the subsequent "hardness" result that arises as a consequence of allowing top-truncated ballots.

Walsh [47] was the first to consider manipulation with single-peaked preferences and he showed that STV remains $\mathcal{NP}$-hard to manipulate for 3 candidates. Subsequently, Faliszewski et al. [29] showed that for many voting protocols which are usually hard to manipulate, restricting the preferences to being single-peaked makes them easy. In particular, they showed that any 3-candidate scoring rule with ($\alpha_1 - \alpha_3$) $\leq 2(\alpha_2 - \alpha_3)$ is easy to manipulate. This result was then extended to obtain a complete characterization for any m-candidate scoring rule by Brandt et al. [9]. Here, we look at 3-candidate scoring rules again and study the impact on complexity of manipulation when top-truncated voting is allowed. We note that the following results for the case of 3-candidate Borda rule was previously shown by Fitzsimmons and Hemaspaandra [33, Theorems 4, 5]. Additionally, since the proofs for Theorem 27 and Theorem 28 in this section are very similar to the ones for Theorem 2 and Theorem 3 in Sect. 4, they have been moved to the appendix.

Theorem 27 *For any 3-candidate scoring rule X that is not isomorphic to plurality or veto, in single-peaked electorates, CWCM with top-truncated votes in $X_\downarrow$ is NP-complete.*

Theorem 28 *For any 3-candidate scoring rule X that is not isomorphic to plurality, in single-peaked electorates, CWCM with top-truncated votes in X_{av} is NP-complete.*

From Theorems 27 and 28, we can see that a relaxation of the complete votes assumption by additionally allowing top-truncated votes actually increases the complexity of CWCM for all 3-candidate scoring rules with ($\alpha_1 - \alpha_3$) $\leq 2(\alpha_2 - \alpha_3)$ (with the exception of veto) from being in $\mathcal{P}$ [29, Theorem 4.4] to being $\mathcal{NP}$-complete when either the round-down or average score evaluation schemes are used. However, with the round-up evaluation scheme manipulation become easy for all m-candidate scoring rules as shown below.

Theorem 29 *In single-peaked electorates, computing if a coalition of manipulators can manipulate plurality$_\downarrow$, veto$_\downarrow$, plurality$_{av}$, and $X_\uparrow$, for any scoring rule X, with weighted top-truncated votes takes polynomial time (for any number of candidates).*

Proof For all the voting rules except plurality$_\downarrow$, the manipulators can simply check if all of them voting p alone will make it a winner. If not they cannot make p a winner. In case of plurality$_\downarrow$, all the manipulators can vote for any single-peaked order consistent with the societal order L that has p at the top.   □

Another interesting point to note here is that Theorems 27, 28, and 29 together also imply that the restriction of preferences to being single-peaked has no effect on the complexity of manipulation with top-truncated ballots, since, as we have already seen, the same corresponding results were obtained in Sect. 4.

Next, we look at CWCM in Copeland$^\alpha$. Yang [49, Theorem 4] showed that in single-peaked electorates CWCM with complete votes is in $\mathcal{P}$ for m-candidate Copeland$^\alpha$, where $m \geq 3$, and below we show that it is $\mathcal{NP}$-complete for even three candidates when top-truncated votes are allowed.

Theorem 30 *In single-peaked electorates, for 3-candidate Copeland$^\alpha$, $\alpha \in \mathbb{Q}, 0 \leq \alpha < 1$, CWCM with top-truncated votes is $\mathcal{NP}$-complete.*

Proof It is easy to see that the problem is in $\mathcal{NP}$. To prove $\mathcal{NP}$-hardness, we show a reduction from the Fixed-difference Subset Sum problem.

Given an arbitrary instance of the Fixed-difference Subset Sum problem, construct the following instance of CWCM, where a, b, and p are the three candidates. Let $a \, L \, p \, L \, b$ be the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S let there be a vote of weight $6K$ voting $(a > b \sim p)$ and $(b > a \sim p)$ each, and a vote of weight $2K$ voting $(p > a > b)$. In T, let each k_i have a vote of weight $2k_i$.

Suppose there exists S_1, S_2 such that $\sum S_1 - \sum S_2 = K$. In Copeland$^\alpha$, it can be assumed that all the manipulators rank p first. So, let the manipulators in S_1 vote $(p > b > a)$, those in S_2 vote $(p > a > b)$, and let the rest vote $(p > a \sim b)$. If $N_V(r, s)$ denotes the total number of votes in V which rank r prior to s, and $D_V(r, s) = N_V(r, s) - N_V(s, r)$, then $D_{S \cup T}(p, a) = 0$ and $D_{S \cup T}(p, b) = 0$. Therefore, the score of p, $s(p) = 2\alpha$. Also since $\sum S_1 - \sum S_2 = K$, $D_T(a, b) = -2K$, while $D_S(a, b) = 2K$. Therefore, $D_{S \cup T}(a, b) = 0$ and so, both receive a score 2α. As a result, p wins by tie-breaking.

Conversely, suppose there exists a successful manipulation in favour of p. If x, y, and z, denote the sum of k_i's of the manipulators in T who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then without taking into account the pairwise election between a and b in T, the score of p, a, and b is 2α, $1 + \alpha$, and α, respectively. Now since $2\alpha < 1 + \alpha$ for all rational $\alpha \in [0, 1)$, therefore, the only way p would win this is if including the pairwise election between a and b in T results in a tie between them. So this implies that $D_{S \cup T}(a, b) = 2K + 2x - 2y = 0$, or $y - x = K$.   □

Our final result for manipulation under single-peaked preferences is the very interesting case of eliminate(Veto).

Theorem 31 *In single-peaked electorates, and in the unique winner model, for eliminate(Veto),*

1. *CWCM with complete votes is in $\mathcal{P}$ when the number of candidates is bounded.*

2. *CWCM with top-truncated votes is $\mathcal{NP}$-complete even for three candidates.*

Proof We first present the polynomial time algorithm for CWCM with complete votes.

Suppose there was an arbitrary set W of the manipulators' votes which—along with S—resulted in p winning. Let the corresponding elimination order be $e = (c_1, \ldots, c_m = p)$. To prove this theorem, we need the following lemmas, the proofs of which are given in the appendix.

Lemma 3 *During each round of the eliminate(Veto), only the rightmost or the leftmost candidate in the linear ordering L is eliminated.*

Lemma 4 *The reverse of e is a single-peaked order with respect to the given linear order L.*

Proof of Theorem 31 *cont'd.* Now since the reverse of every elimination order e caused by any arbitrary manipulation in favour of p is single-peaked (Lemma 4), the manipulators can try out all possible single-peaked orders which have p at the top and can try to induce e by collectively voting for the same. This is enough because we know that in eliminate(Veto) any elimination order that is achieved by an arbitrary set of manipulators' votes can be achieved if all the manipulators vote the same way (by using the reverse of the elimination order) [12, Lemma 12]. And since, when the number of candidates is bounded, there are only polynomial number of single-peaked orders which have p at the top, this can be done in polynomial time.

We now go on to show the hardness of CWCM with top-truncated votes for 3-candidate eliminate(Veto). Clearly, the problem is in $\mathcal{NP}$. To show $\mathcal{NP}$-hardness, we show a reduction from the Partition problem.

Given an arbitrary instance of the Partition problem, construct the following instance of CWCM, where a, b, and p are the three candidates. Let $a \, L \, b \, L \, p$ be the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(a > b > p), (b > a > p), (b > p > a), (p > b > a), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S let there be a voter of weight $K + 2$ voting $(b > p > a)$, voter of weight $K - 1$ voting $(b > a > p)$, a voter of weight 1 voting $(a > b > p)$, a voter of weight 2 voting $(p > a \sim b)$, and a voter of weight 3 voting $(a > b \sim p)$. In T let each k_i have a vote of weight k_i.

The first thing to observe is, p cannot be unique winner if a is eliminated in the first round. This is because once a is eliminated, $score(b) = score(p) = 2K + 2$. Therefore, for p to be a unique winner, b has to be eliminated in the first round. This in turn implies that the total score of a and p should be at least 1 more than that of b (since we are in the unique winner model). So we have,

$$score_T(a) + K + 3 \geq 2K + 3 + score_T(b) \implies score_T(a) - score_T(b) \geq K$$
$$score_T(p) + K + 4 \geq 2K + 3 + score_T(b) \implies score_T(p) - score_T(b) \geq K - 1,$$

where $score_T(p)$ is the score p receives from the votes in T.

Now, it is possible to show that the above equations can be satisfied only if b receives a score of 0 from T. Therefore, this implies that all the votes are concentrated on $(p > a \sim b)$ and $(a > b \sim p)$. As a result, given the equations above, we basically have two possibilities: a total weight of $K + 1$ voting for $(a > b \sim p)$, $K - 1$ voting $(p > a \sim b)$, or a total weight of K voting $(a > b \sim p)$ and $(p > a \sim b)$ each. In the first case it is easy to see that p will be eliminated in the second round, while in the latter case p will be a unique winner. Hence p can be a unique winner if and only if there exists a partition. □

Theorem 31 is most interesting not because of the fact that it follows our theme of top-truncated voting reinstating combinatorial protections, but for the following other reasons.

First is the very unusual behaviour that it is showing. Eliminate(Veto), when there are only a bounded number of candidates and in the unique winner model, is known to be in $\mathcal{P}$ in most settings—from CWCM with complete votes in the general case [12], to CWCM with top-truncated votes in the general case (see Corollary 5 in Sect. 4), and even when there is only partial information (in the form of top-truncated votes) on the non-manipulators' votes (see Theorem 22 in Sect. 4 and the discussion following that). However, here, with single-peaked preferences and with top-truncated votes, it is $\mathcal{NP}$-complete even when there are only three candidates. Second, what makes Theorem 31 even more interesting is the fact that this actually serves as a counterexample (with the only caveat that, to be fair, they had considered only complete votes in that paper) to a conjecture stated by Faliszewski et al. [29] where they say that they do not expect the complexity of manipulation for "any existing, natural voting system" to increase when moving from the general case (where there is no restriction on the preferences) to the single-peaked case. But this is exactly what we are seeing here.

5.3 Bribery in single-peaked electorates

Faliszewski et al. [25] were the first to look at the complexity of bribery in elections. Subsequently, the problem was studied by Brandt et al. [9] in single-peaked settings and there they showed that many of the combinatorial protections for bribery vanish when the preferences are restricted to being single-peaked. Here, we revisit the problem of bribery in single-peaked settings and we see if bribery too, like manipulation, fits into our theme of reinstating combinatorial protections in single-peaked elections through top-truncated voting.

5.3.1 Weighted-bribery in scoring rules

Here, we first derive the results for 3-candidate scoring rules in single-peaked settings when only complete votes are allowed. Subsequently, we do the same when top-truncated ballots are allowed. The $\mathcal{NP}$-completeness proofs here use an idea that is similar to the one used by Faliszewski et al. [25] in Theorem 4.9, where they use a reduction from a modified version of the weighted manipulation problem to show that α-weighted-bribery is $\mathcal{NP}$-complete when it isn't the case that $\alpha_2 = \alpha_3 = \cdots = \alpha_m$. Although it is possible to extend the results of the complete votes case here to any m-candidate scoring rule, thanks to the complete characterization of weighted manipulation for scoring rules given by Brandt et al. [9], we stick to the 3-candidate case here because, to the best our knowledge, such a complete characterization is not yet known when top-truncated ballots are allowed.

Let us first define the modified version of manipulation that we will use to reduce to the problem of weighted-bribery. The modified problem defined here is similar to the one used by Faliszewski et al. [25], with the only difference that in their problem all the manipulators need to have weights at least twice as much as the weight of the heaviest non-manipulator, while in our case we require that all the manipulators need to have weights at least thrice as much as the weight of the heaviest non-manipulator.

Definition 12 (*CWCM'*) CWCM$'$ is the same problem as CWCM with the restriction that each manipulative voter has a weight at least thrice as much as the weight of the heaviest non-manipulator.

Theorem 32 *In single-peaked electorates, CWCM$'$ with complete votes is $\mathcal{NP}$-complete for 3-candidate scoring rules when $(\alpha_1 - \alpha_3) > 2(\alpha_2 - \alpha_3)$.*

Proof To prove the above theorem for the complete votes case, we make use of Faliszewski et al. [29]'s proof of Theorem 4.4 which shows how CWCM is $\mathcal{NP}$-complete for 3-candidate scoring rules when $(\alpha_1 - \alpha_3) > 2(\alpha_2 - \alpha_3)$. The first observation to make here is that the reduction in [29, Theorem 4.4] works even if we use Partition'. So, to prove the $\mathcal{NP}$-completeness of CWCM', the only thing we need to do is to make sure that the manipulators used in the reduction have weights at least thrice as that of the heaviest non-manipulator. This in turn can be ensured if we can somehow replace every non-manipulator who has a weight greater than the threshold (which here is one-third the weight of the lightest manipulator) with several non-manipulators each of whose weight is less than the threshold.

Again, in the proof of [29, Theorem 4.4], if we use Partition' for reduction then there will be $2n$ non-manipulators (instead of just 2 in the original proof) each of weight $3(2\alpha_1 - \alpha_2)K$ such that n of them vote for $(a > p > b)$ and the other n vote $(b > p > a)$. Additionally, each of manipulators will now have a weight of $3(\alpha_1 - 2\alpha_2)a_i$, where $\{a_1, \ldots, a_n\}$ is an arbitrary instance of Partition' with $a_i \geq K, \forall i$ and $\sum_i a_i = 2nK$ (the rationale behind multiplying an additional factor of 3 to the weights of each manipulator and non-manipulator will be evident below). But now, since we need the weights of the manipulators to be at least thrice as much as the weight of the heaviest non-manipulator, we will have to replace each non-manipulator by at most

$$\left\lceil \frac{3(2\alpha_1 - \alpha_2)K}{\left\lfloor \frac{1}{3}(3(\alpha_1 - 2\alpha_2))a_{\min} \right\rfloor} \right\rceil$$

non-manipulators, where $a_{\min} = \operatorname{argmin}_i\{a_i\}$. Since $a_{\min} \geq K$, and $(\alpha_1 - 2\alpha_2) \geq 1$, we have,

$$\left\lceil \frac{3(2\alpha_1 - \alpha_2)K}{\left\lfloor \frac{1}{3}(3(\alpha_1 - 2\alpha_2))a_{\min} \right\rfloor} \right\rceil \leq \left\lceil \frac{3(2\alpha_1 - \alpha_2)K}{a_{\min}} \right\rceil \leq \lceil B \rceil$$

where $B = 3(2\alpha_1 - \alpha_2)$ is a constant. Now, since this splitting can be done in polynomial time, that completes the proof of the theorem.   □

We now show the result for weighted-bribery in scoring rules.

Theorem 33 *In single-peaked settings, weighted-bribery with complete votes is in $\mathcal{P}$ for 3-candidate scoring rules when $(\alpha_1 - \alpha_3) \leq 2(\alpha_2 - \alpha_3)$ and is $\mathcal{NP}$-complete otherwise.*

Proof Let a, b, and p be the three candidates. Without loss of generality we can assume that $\alpha_3 = 0$. We will discuss both cases ($\alpha_1 \leq 2\alpha_2$ and $\alpha_1 > 2\alpha_2$) separately.
Case 1 $\alpha_1 \leq 2\alpha_2$. Here we basically have two cases: i) when the input linear ordering of the candidates is $a \, L \, b \, L \, p$ and ii) when the input linear ordering of the candidates is $a \, L \, p \, L \, b$. In scoring rules, it is reasonable to assume that the bribed voters will always be made to vote in such a way that p will be at the top. Therefore, in the first case, since there is only one vote $(p > b > a)$ that has p placed at the top, all the bribed voters will be bribed to vote $(p > b > a)$. So now the only part remaining part here is to identify who to bribe. This can be done in the following way. Let us suppose that there was a successful bribery for a given instance. Since the only allowed votes for $a \, L \, b \, L \, p$ are $\{(a > b > p), (b > a > p), (b > p > a), (p > b > a)\}$, any successful bribery would have resulted in bribing x_1 voters initially voting $(a > b > p)$, x_2 voters initially voting $(b > a > p)$, and x_3 voters initially voting $(b > p > a)$. Now, to identify these x_1, x_2, and x_3, we simply need to iterate through all the feasible (a solution (x_1, x_2, x_3) is feasible if the input set of votes has at least x_1 voters voting $(a > b > p)$, at least x_2 voters voting $(b > a > p)$, and at least x_3 voters voting $(b > p > a)$) integral solutions of $x_1 + x_2 + x_3 \leq k$, where k is the bribe limit, pick

the x_1 heaviest voters voting ($a > b > p$), the x_2 heaviest voters voting ($b > a > p$), the x_3 heaviest voters voting ($b > p > a$), bribe all of them to vote ($p > b > a$), and check if it makes p a winner. If yes, we accept. Else, we continue on to the next set of solutions which satisfy the equation. Since there are only $O(k^3)$ such solutions, and since $k \le n$, this can be done in polynomial time.

For the case $a \, L \, p \, L \, b$, the only allowed votes are $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a)\}$, and so we can bribe a voter to vote either ($p > a > b$) or ($p > b > a$). Therefore, the main task here is to first identify the right voters to bribe and then decide on whether to make them vote ($p > a > b$) or ($p > b > a$). To do this, let us assume there was a successful bribery in favour of p. This in turn would have resulted in bribing x_1 voters initially voting ($a > p > b$), x_2 voters initially voting ($b > p > a$), x_3 voters initially voting ($p > a > b$), and x_4 voters initially voting ($p > b > a$). Now, let $F = \{(x_1, x_2, x_3, x_4) \mid x_1 + x_2 + x_3 + x_4 \le k$ and (x_1, x_2, x_3, x_4) is feasible$\}$. For each $x = (x_1, x_2, x_3, x_4) \in F$, pick the heaviest x_1 voters voting ($a > p > b$), the heaviest x_2 voters voting ($b > p > a$), the heaviest x_3 voters voting ($p > a > b$), and the heaviest x_4 voters voting ($p > b > a$). Let S' be the set of all these voters. Next, calculate the scores of p, a, and b considering all the votes in $S - S'$, where S is the set of all votes given as part of the input. Let these scores be denoted by $s_{S-S'}(p), s_{S-S'}(a)$, and $s_{S-S'}(b)$ respectively. Now for the case $aLpLb$, since $\alpha_1 \le 2\alpha_2$, for any set of votes, p cannot lose to both a and b. Therefore the only possible cases for the scores $s_{S-S'}(p), s_{S-S'}(a)$, and $s_{S-S'}(b)$ are the following. For each of them we mention how the bribery needs to be done.

1. $s_{S-S'}(p) \ge s_{S-S'}(a)$, $s_{S-S'}(p) \ge s_{S-S'}(b)$: In this case, for each voter in S', make him or her vote either ($p > a > b$) or ($p > b > a$), if they aren't already voting for either of them.
2. $s_{S-S'}(b) > s_{S-S'}(p)$, $s_{S-S'}(a) \le s_{S-S'}(p)$: In this case, for each voter in S', make him or her vote ($p > a > b$), if they aren't already.
3. $s_{S-S'}(a) > s_{S-S'}(p)$, $s_{S-S'}(b) \le s_{S-S'}(p)$: Here, for each voter in S', make him or her vote ($p > b > a$), if they aren't already.

Doing the above takes only polynomial time since there are only $O(k^4)$ feasible solutions and $k \le n$.

Case 2 $\alpha_1 > 2\alpha_2$. To show that weighted-bribery in this case is $\mathcal{NP}$-complete, we use a reduction from CWCM$'$. Note that to show this we use the linear ordering $a \, L \, p \, L \, b$ since there are polynomial time algorithms for both CWCM$'$ and weighted-bribery when p is at one of the ends in the societal order. Now, given an arbitrary instance (C, S, T, p) of CWCM$'$, we construct the following instance (C, V, p, k) where C is set of candidates, $V = S \cup T'$ where T' is the set of voters from T voting either ($a > p > b$) or ($b > p > a$). We set $k = |T|$.

It is easy to see that if a successful manipulation exists then bribery with at most $k = |T|$ bribes is possible. To prove the converse we basically need to show that any bribery with at most $|T|$ bribes is possible if we bribe only the voters from T'. For this, let us assume this were not the case and that there was a voter $v \in V - T'$ who was part of the successful bribery in favour of p. By bribing v, the gain for p against a or b is at most $(\alpha_1 + (\alpha_1 - \alpha_2))w(v)$, where $w(v)$ is the weight of voter v (this is so because, p can gain at most $(\alpha_1 - \alpha_2)$ when it moved to the top since in any valid single-peaked order according to L it is placed in the second position; the other α_1 term comes from the fact that candidate a or b can lose at most α_1 in the worst-case as a result of this bribery). On the other hand if a voter $v' \in T'$ was bribed, then the gain for p against a or b is at least $(\alpha_1 - \alpha_2)w(v')$. But then, according to the restriction imposed on CWCM$'$, any $v' \in T'$ has a weight at least thrice as much as

the heaviest voter in $V' - T$. This in turn implies that $w(v') \geq 3w(v)$, and so we might as well bribe v' instead of v. As a result, any bribery in favour of p can be achieved by bribing only the voters from T'. Therefore, existence of a successful bribery implies that a successful manipulation exists in CWCM$'$.   □

For the case of top-truncated ballots, we proved in Theorem 27 that, in single-peaked settings, CWCM with top-truncated votes is $\mathcal{NP}$-complete for all 3-cand-idate scoring rules except plurality and veto when the evaluation scheme was round-down. Similarly, we also showed in Theorem 28 that CWCM with top-truncated votes is $\mathcal{NP}$-complete for all 3-candidate scoring rules except plurality when the evaluation scheme was average-score. Based on these two theorems it is easy to see that we can make similar 'splitting' arguments as in the proof of Theorem 32 to prove these results hold true even for CWCM$'$. As a result, we can state the following results which can be proved by using a reduction from CWCM$'$ similar to the one shown in case 2 of Theorem 33.

Theorem 34 *For any 3-candidate scoring rule X that is not isomorphic to plurality or veto, in single-peaked electorates, weighted-bribery with top-truncated votes in $X_\downarrow$ is $\mathcal{NP}$-complete.*

Theorem 35 *For any 3-candidate scoring rule X that is not isomorphic to plurality, in single-peaked electorates, weighted-bribery with top-truncated votes in X_{av} is $\mathcal{NP}$-complete.*

5.3.2 Weighted-bribery in eliminate(Veto)

Here we look at the problem of weighted-bribery in eliminate(Veto) in single-peaked electorates and yet again observe that allowing top-truncated voting increases the complexity of weighted-bribery from being in $\mathcal{P}$ to being $\mathcal{NP}$-complete.

Theorem 36 *In single-peaked electorates, and in the unique winner model, for 3-candidate eliminate(Veto),*

1. *weighted-bribery with complete votes is in $\mathcal{P}$.*
2. *weighted-bribery with top-truncated votes is $\mathcal{NP}$-complete.*

Proof First, we discuss the polynomial time algorithm for weighted-bribery when only complete votes are allowed.

Let a, b, and p be the three candidates. Without loss of generality, we need to consider only two linear orders: $a \ L \ b \ L \ p$ and $a \ L \ p \ L \ b$.

(i) $a \ L \ b \ L \ p$: For the given linear ordering, since b cannot be eliminated in the first round as there is no valid preference order according to L in which it can be placed last, only the following elimination orders are possible: $\{(a, b, p), (a, p, b), (p, a, b), (p, b, a)\}$, where (a, b, p) implies that a is eliminated in the first round, followed by b, and then p (which is the winner). We will consider each of these cases separately.

Case 1 (a, b, p). Here no bribery is required as p is already the winner.

Case 2 (a, p, b). In this case, it is easy to see that a simple greedy strategy would suffice. Since b cannot be eliminated in the first round (as there is no vote with b at the end according to L), we only need to make sure that p wins against b in the second round. This can be ensured by doing the following:

Starting with the heaviest voter among all voters who do not rank p above b, bribe him or her to vote $(p > b > a)$. If p is the winner after the bribe, then accept. If not, continue the same until number of voters bribed exceeds the limit k. If the limit exceeds, reject.

Case 3 (p, a, b). Here for p to win, it has to be saved from elimination in the first round, and subsequently has to beat b in the second round. To do this, we can proceed in two phases: In the first phase, starting with the heaviest voter among all voters who do not rank a last, bribe them to vote $(p > b > a)$. After each bribe, check if $LV(a) > LV(p)$, where $LV(a)$ denotes the total weight of all the voters who have placed a last in their preference ordering. Once that is done, i.e., once $LV(a) > LV(p)$, we check if p is the winner. If yes, we accept. Else, we move on to the second phase where starting from the heaviest voter but now among all voters who do not rank p above b, we bribe them to vote $(p > b > a)$. Here again, after each bribe, we see if it results in p being the winner. If it does, we accept. Otherwise, if at any point during this algorithm the number of voters bribed exceeds the limit k, then we reject.

Case 4 (p, b, a). Follow the exact same algorithm as in case 3.

Before considering the second linear order, we need the following lemma, the proof of which is given in the appendix.

Lemma 5 *For the linear order $a \ L \ p \ L \ b$, when the input elimination order is (a, p, b), if there exists a successful bribery, then it can be induced by the elimination order (a, b, p) (or in other words, we do not need to consider any other elimination orders).*

(ii) $a \ L \ p \ L \ b$: Here, only the following elimination orders are possible: $\{(a, b, p), (a, p, b), (b, a, p), (b, p, a)\}$. Again, we will consider each of these cases separately.

Case 1 $(a, b, p), (b, a, p)$. In both these cases no bribery is required as p is already the winner.

Case 2 (a, p, b). Here the situation is a little more complicated since the voters can be bribed to either induce the elimination order (a, b, p) or (b, a, p). But although this is the case, in Lemma 5 we prove that inducing (a, b, p) is enough as inducing the other would require bribing more voters. So now, this case is similar to case 2 for the linear order $a \ L \ b \ L \ p$ and we can use the use the algorithm described there.

Case 3 (b, p, a). This is similar to case 2 above with a and b interchanged.

This concludes the proof of the first part.

To prove the second part, we show how Partition′ can be used to prove that weighted-bribery is hard for eliminate(Veto) when top-truncated votes are allowed. The proof essentially follows an idea similar to the one used for proving the corresponding result for manipulation in Theorem 31, only that we show this reduction through the modified partition problem. The main motivation behind using the modified partition problem is to basically facilitate a situation that will enable us to argue that the bribery can be restricted to a certain set of voters (who correspond to the manipulators in the corresponding manipulation instance).

Now, it is easy to see that the problem is clearly in $\mathcal{NP}$. To prove $\mathcal{NP}$-hardness, we show a reduction from Partition′. Given an arbitrary instance $\{a_1, \ldots, a_n\}$ of Partition′, where $\sum_i a_i = 2nK$ and $a_i \geq K, \forall i$, construct an instance of weighted-bribery (C, V, p, k), where $C = \{a, b, p\}$ is the set of candidates, $a \ L \ b \ L \ p$ is the linear order of the candidates, and V is the set of the following voters.

1. For each a_i, construct a voter v_i whose weight is a_i and who votes for $(b > a > p)$. Let T be the set of all these voters.
2. Construct the following set of voters S: n voters of weight K each voting $(b > p > a)$, n voters of weight $K - 1$ each voting $(b > a > p)$, $n - 1$ voters of weight 1 each voting for $(b > a > p)$, 1 voter of weight 1 voting $(a > b > p)$, 1 voter of weight 2 voting

$(b > p > a)$, 1 voter of weight 2 voting $(p > a \sim b)$, and 1 voter of weight 3 voting $(a > b \sim p)$.

Set the bribe limit $k = n$ and $V = S \cup T$.

First thing to notice is that, if a is eliminated in the first round then p cannot be a unique winner, no matter how we choose the n voters to bribe. Therefore the only way to make p a unique winner is by making sure that a does not get eliminated. Now since the bribe limit is n, the maximum total weight of the voters bribed cannot exceed $2nK$. Also, using arguments similar to the one used in the proof of Theorem 31, we can see that any bribery with at most n bribes is possible only if there is an equal weight of nK that is assigned to both $(p > a \sim b)$ and $(a > b \sim p)$. Therefore, the only part that remains to be argued is to show that this bribery with at most n bribes can be achieved if we bribe only those voters in T.

Before we show this, let us see the objectives involved in the bribery of this instance. First is that any bribe results in changing the vote of the corresponding voter to either $(p > a \sim b)$ or $(a > b \sim p)$. Secondly, while bribing a voter, we always want the gain for p against every other candidate to be as much as possible. And lastly, we also need to ensure that while a is given enough score so that it is not eliminated in the first round, any bribe to vote $(a > b \sim p)$ does not result in the score of p decreasing (because if the score of p decreases as a result of the bribery, then p cannot become a unique winner since the difference between the scores of p and a when considering only the votes from S is just one).

Now, having seen the objectives, let us suppose that there was a voter $v \in S$ who was part of a successful bribery in favour of p. Before the bribe, v has one among $(b > p > a)$, $(b > a > p)$, $(a > b > p)$, $(p > a \sim b)$, and $(a > b \sim p)$ has his or her vote. But given the objectives, the only rational choice of a voter who has to be bribed is one who's current vote is $(b > a > p)$ (bribing one with $(a > b \sim p)$ is irrational since to compensate the reduction in the score of a we will have to bribe another voter; bribing a voter with $(b > p > a)$ is irrational since if this voter was bribed to vote $(p > a \sim b)$ then the gain for p would have been more if instead a voter voting $(b > a > p)$ would have been bribed, or otherwise if he or she was bribed to vote $(a > b \sim p)$ then another voter would have to be bribed to compensate the reduction in p's score; bribing a voter with $(a > b > p)$ is irrational since bribing this voter has the same effect as bribing a voter with $(b > a > p)$ and the weights of all the voters voting $(b > a > p)$ is at least as much as the weight of the voter voting $(a > b > p)$). But then, all the voters in T are not only voting $(b > a > p)$ initially, but they also have weights which are at least as much as of those voters voting $(b > a > p)$ in S. Hence, any bribery that is possible with at most n bribes can be achieved if we bribe only the voters from T. Now, since the total weight of all voters in T is equal to $2nK$, this implies that p is a unique winner if and only if there is a partition. $\square$

5.3.3 Is weighted-bribery for weak-Condorcet consistent rules always easy?

Brandt et al. [9] showed that in single-peaked electorates weighted-bribery is in $\mathcal{P}$ for all weak-Condorcet consistent voting rules (see [9, Theorem 4.4] for a more general result). The polynomial results in their theorem and the reason why it was possible to consider all weak-Condorcet consistent voting rules together was because of the well-known property of single-peaked electorates where it is guaranteed that there is always at least one weak-Condorcet winner (the top choices of the "median" voters are always weak-Condorcet winners). However, this property no longer holds when top-truncated votes are allowed. As has also been pointed out by Cantala [11], it is not even necessary that a weak-Condorcet winner exists in such settings. We illustrate this with the following example.

Example 3 Let $C = \{a, b, c, d, e\}$ with $a \ L \ b \ L \ c \ L \ d \ L \ e$ as the linear ordering. Let there be 5 voters $\{v_1, \ldots, v_5\}$ with and following votes, respectively: $(a > b > c > d \sim e)$, $(b > c > d \sim e \sim a)$, $(c > d > e \sim b \sim a)$, $(d > e > a \sim b \sim c)$, and $(e > d > c \sim b \sim a)$. Now, it is easy to see that in the pairwise majority relation, a and b lose to d, c loses to b, and both d and e lose to c. Since everyone loses at least once, there is no weak-Condorcet winner.

Because of the above we can no longer consider all weak-Condorcet consistent rules together as done by Brandt et al. [9] and exploit the connection between the weak-Condorcet winner(s) and "median" voters to come up with polynomial time algorithms for weighted-bribery. In fact, next we show that for 3-candidate Baldwin's rule (also known as Fishburn's version of Nanson's rule [44]), which is a weak-Condorcet consistent rule in single-peaked electorates [9], weighted-bribery is $\mathcal{NP}$-complete when top-truncated ballots are allowed. To show this we use an idea similar to the one used in Theorem 36. The complete proof can be found in the appendix.

Theorem 37 *In single-peaked electorates, weighted-bribery with top-truncated votes is NP-complete for 3-candidate Baldwin's rule.*

5.4 Does top-truncated voting always increase the complexity of manipulation in SP electorates?

Although we have seen instances, like in 3-candidate scoring rules with round-up evaluation scheme, where the complexity of manipulation decreases as a result of moving from a purely single-peaked setting to a setting where top-truncated votes are allowed, we haven't really seen examples of any other voting rule which shows this behaviour. Moreover, we also know that with a different evaluation scheme like round-down or average-score this behaviour is no longer seen for even 3-candidate scoring rules. Therefore, a natural question one could ask is: "What role does the evaluation scheme play? Is it possible that given a voting rule one can always construct an evaluation scheme so that it will result in an increase in the complexity of manipulation when top-truncated voting is allowed in single-peaked electorates?". Alternatively, one could also ask: "Is there a voting system for which it is always easy to manipulate when top orders are allowed?". We answer the former question in the negative and the latter one in the affirmative. We show that, as long as all the unranked candidates are assumed to be tied and are assumed to be ranked below the ranked candidates (which is the natural definition of a top-truncated vote), there is at least one voting system for which, irrespective of how the top-truncated votes are dealt with, it is $\mathcal{NP}$-hard to manipulate in purely single-peaked settings, but is easy to manipulate when top-truncated votes are allowed.

Theorem 38 *There exists a voting system for which, in single-peaked settings,*

1. *CWCM with complete votes is $\mathcal{NP}$-complete.*
2. *CWCM with top-truncated votes is in P.*

Proof Let us first define the (artificial) voting system that we consider here.

Definition 13 (*Artificial voting rule (AVR)*) Given the set, V, of voter preferences, the score of each candidate $c \in C$ is calculated as

$$s(c) = \sum_{v \in V} \sum_{\substack{a \in C \\ a \neq c \\ c >_v a}} (m - pos(c))w(v)$$

where $pos(c)$ is the position of candidate c in the preference order of v if it is ranked ($pos(c) = i$ if c is ranked in the ith position by v) and is m otherwise, $m = |C|$, $w(v)$ is the weight of voter v, and $c >_v a$ denotes that c is ranked above a by v.

Example 4 Let $C = \{a, b, c, d, e\}$ be the set of candidates and let there be a single voter of weight w with the preference ordering ($d > b > a > e > c$). Then the scores of the candidates are $s(a) = 4w$, $s(b) = 9w$, $s(c) = 0$, $s(d) = 16w$, and $s(e) = w$.

So now we need to show that for AVR, in single-peaked settings, CWCM with complete votes is $\mathcal{NP}$-complete while CWCM with top-truncated votes is in $\mathcal{P}$.

We show the first part by a reduction from Partition. Given an arbitrary instance $\{k_i\}_{1 \le i \le t}$, $\sum_i k_i = 2K$, of Partition, construct the following instance of CWCM, where a, b, and p are the three candidates and $a \ L \ p \ L \ b$ is the linear ordering over the candidates. In S let there be two voters of weight $7K$ each voting ($a > p > b$) and ($b > p > a$) respectively, and two voters of weight K each voting ($p > a > b$) and ($p > b > a$) respectively. In T let each k_i have a vote of weight k_i. According to the rule defined above, the scores of a, b, and p are $29K$, $29K$, and $22K$ respectively.

Suppose there exists a partition. Let those manipulators in one half vote ($p > a > b$) while those in the other half vote ($p > b > a$). As a result the scores of a, b, and p are all $30K$ and hence p is a winner.

Conversely, suppose there exists a manipulation in favour of p. In AVR it is reasonable to assume that all the manipulators place p first. So, now, let x and y be total weight of the manipulators in T who vote ($p > a > b$) and ($p > b > a$) respectively. Since there exists a successful manipulation in favour of p, the score of p should be at least as much as that of a. Therefore, we have: $22K + 4x + 4y \ge 29K + x$. Using the fact that $x + y = 2K$, this simplifies to $x \le K$. Doing the same with respect to p and b we have, $y \le K$. But then since $x + y = 2K$, this implies that $x = K$ and $y = K$ and that there exists a partition.

For the second part, it is easy to see that the optimal strategy for the manipulators is to just vote ($p > a \sim b$). □

6 Conclusions and future work

While there have been many previous works which have studied the complexity of different manipulative actions, a disproportionate number of them have been under the assumption that the agents specify complete preference orderings over the set of candidates. Although it is an interesting setting to study, there are many practical situations where the agents may not be willing or may simply not be able to provide this information. It is these kind of scenarios that we study here, and in particular, we focus on situations where the agents are allowed to specify partial preferences in the form of top-truncated ballots.

In looking at elections with top-truncated ballots, we addressed two related questions. First was the question of what the impact on complexity of manipulation is when top-truncated ballots are allowed. This was discussed in Sect. 4 and here we studied the problem of manipulation—both constructive and destructive—in weighted elections when the agents are allowed to specify top-truncated preferences and also looked at the impact on manipulation when there is uncertainty about the non-manipulators' votes. We devoted most of this section studying the first problem—i.e., when the manipulators have complete information on the non-manipulators—and the results for the same are summarized in Table 1. Subsequently, we also explored the second avenue i.e., when there is uncertainty about the non-manipulators' votes. Here we discussed two possible ways in which the uncertainty can be modeled and

Table 1 Complexity of CWCM and DWCM with top-truncated votes

Voting rule	CWCM (#cand)	DWCM (#cand)
$X_\uparrow$	$\mathcal{P}$	$\mathcal{P}$
Plurality$_\downarrow$	$\mathcal{P}$	$\mathcal{P}$
Veto$_\downarrow$	$\mathcal{P}$	$\mathcal{P}$
$X^1_\downarrow$	$\mathcal{NP}$-complete (3)	$\mathcal{P}$
Plurality$_{av}$	$\mathcal{P}$	$\mathcal{P}$
X^2_{av}	$\mathcal{NP}$-complete (3)	$\mathcal{P}$
eliminate(Veto)	$\mathcal{P}$	$\mathcal{P}$
eliminate(X^3)	$\mathcal{NP}$-complete (3)	$\mathcal{NP}$-complete (3)
Plurality with runoff	$\mathcal{NP}$-complete (3)	$\mathcal{NP}$-complete (3)
$Copeland^\alpha$, $\alpha \in [0, 1)$	$\mathcal{NP}$-complete (3)	$\mathcal{P}$
Maximin	$\mathcal{P}$	$\mathcal{P}$

The entries in bold indicate that there is a change in complexity of CWCM as compared to the case of complete votes, the non-highlighted ones indicate that there is no change to the worst-case complexity as compared to the case of complete votes, and the italicized entries indicate those rules for which there is more opportunity for manipulation but for which the worst-case complexity is still the same as compared to the case of complete votes
X: all scoring rules
X^1: all scoring rules except plurality and veto
X^2: all scoring rules except plurality
X^3: all scoring rules except veto

Table 2 Change in complexity of CWCM when moving from complete votes to top-truncated votes in single-peaked electorates

Single-peaked electorates			
Voting rule	CWCM (complete orders)	Voting rule	CWCM (top-t orders)
3-X	$\mathcal{P}$, if ($\alpha_1 \leq 2\alpha_2$) [29]	3-$X^1_\downarrow$	$\mathcal{NP}$-complete
3-X	$\mathcal{NP}$-complete, if ($\alpha_1 > 2\alpha_2$) [29]	3-X^2_{av}	$\mathcal{NP}$-complete
eliminate(Veto)	$\mathcal{P}$	eliminate(Veto)	$\mathcal{NP}$-complete
m-Copeland$^\alpha$ $\alpha \in [0, 1), m \geq 3$	$\mathcal{P}$ [49]	3-Copeland$^\alpha$, $\alpha \in [0, 1)$	$\mathcal{NP}$-complete

X : all scoring rules
X^1 : all scoring rules except plurality and veto
X^2 : all scoring rules except plurality
X^3 : all scoring rules except veto

we also showed that in both cases even individual manipulation under uncertainty was hard when constructive coalitional manipulation was hard.

The second issue we looked at is the question of what the impact of top-truncated ballots are on the complexity of manipulation and bribery when the electorate has structured preferences. In particular, we focused on single-peaked preferences, and we observed a number of surprising results which in turn formed the theme of Sect. 5—of reinstating combinatorial protections in single-peaked electorates by allowing top-truncated voting. We observed this behaviour first in the case of manipulation and showed how for different voting protocols manipulation with complete votes was in $\mathcal{P}$ whereas manipulation with top-truncated votes jumped to being $\mathcal{NP}$-complete. These results were followed by the results for bribery where,

Table 3 Change in complexity of weighted-bribery when moving from complete votes to top-truncated votes in single-peaked electorates

Single-peaked electorates			
Voting rule	CWCM (complete orders)	Voting rule	CWCM (top-t orders)
3-X	$\mathcal{P}$, if ($\alpha_1 \leq 2\alpha_2$)	3-$X^1_\downarrow$	$\mathcal{NP}$-complete
3-X	$\mathcal{NP}$-complete, if ($\alpha_1 > 2\alpha_2$)	3-X^2_{av}	$\mathcal{NP}$-complete
eliminate(Veto)	$\mathcal{P}$	eliminate(Veto)	$\mathcal{NP}$-complete
3-Baldwin's rule	$\mathcal{P}$ [9]	3-Baldwin's rule	$\mathcal{NP}$-complete

X : all scoring rules
X^1 : all scoring rules except plurality and veto
X^2 : all scoring rules except plurality
X^3 : all scoring rules except veto

again, we observed similar behaviour for the voting rules considered. In addition to the above results (summarized in Tables 2, 3) we also showed an instance of a natural voting system (eliminate(Veto)) where, contrary to intuition, the complexity of manipulation, when top-truncated ballots are allowed, actually increases from being in $\mathcal{P}$ in the general case to being $\mathcal{NP}$-complete in the single-peaked case. Finally, we concluded our discussion in Sect. 5 by showing the example of a voting system where allowing top-truncated voting isn't beneficial in the sense that it actually results in a decrease in the complexity of manipulation.

There are numerous possible avenues for future work. Foremost would be to consider other types of partial preferences like bottom orders,[3] weak orders etc. Although Fitzsimmons and Hemaspaandra [33] have done some work in this direction, they only look at specific protocols and moreover largely (with the exception of their results for the 3-candidate Borda rule in single-peaked settings) consider only the general setting—i.e., when the preferences of the agents do not have a particular structure. Especially in the case of structured preferences, it would be interesting to see if similar observations as in Sect. 5 can be made. Second, while studying the impact of top-truncated ballots on the complexity of manipulative actions in structured electorates, we have only considered the problems of manipulation and bribery, but not control. Therefore, we feel that it would be worthwhile to see if similar observations can be made for the problem of control as well. Third, while considering structured preferences, we have only considered single-peaked preferences. However, there are papers in the literature that look at the notion of near-single-peakedness (see, e.g., the work of Erdélyi et al. [23]) and so seeing if we can obtain similar results when considering such electorates can be interesting. Fourth, we have considered only weighted elections in this work, but we believe that looking at the unweighted case would be even more insightful as it would help us understand the inherent "hardness" that is embedded in the voting rule itself. Often what seems to be happening in weighted elections is that the computational complexity seems to be arising mainly from the weights themselves, thus denying us the opportunity to understand the voting rules better. Therefore, it is definitely something to be considered as a future research direction. With regards to the impact of partial voting on the complexity of manipulative actions, to the best of our knowledge, there is only one work by Narodytska and Walsh [42] that has some results on unweighted elections. Finally, a possible criticism regarding the results in this work (and in fact any work which considers the same notions) could be that the results are

---

[3] Bottom orders can be useful in scenarios where one is indifferent between some candidates at the top but truly dislike some other candidates; there are several papers in computational social choice that look at bottom orders (see, e.g., [6,29,33]).

in the worst-case and that we use $\mathcal{NP}$-hardness—which in many cases does not necessarily reflect the actual difficulty in practice—as the complexity measure. Therefore, one long term research direction would be to look at the average-case complexity for manipulative actions in elections. There has been some work that has been done in this direction (see, e.g., the work of Friedgut et al. [35]), but we believe that there is still scope for much more.

Appendix: Additional proofs

Proofs for Sect. 3.3

Lemma 1 *Partition′ is NP-complete.*

Proof It is easy to see that Partition′ is in $\mathcal{NP}$. To prove $\mathcal{NP}$-hardness, we show a reduction from the Partition (P1) problem. Let $\{a_1, \ldots, a_n\}$, where $\sum_i a_i = 2M$, be an arbitrary instance of the Partition problem. Next, construct the following instance of Partition′(P2):

1. For each a_i, construct $b_i = a_i + 2M^2$ and $b_i' = 2M^2$.
2. Construct $X_1 = X_2 = 2M^2 n + 4M^2 - M$.

Now, it can be seen that $\sum_i (b_i + b_i') + X_1 + X_2 = 2 \cdot (2n + 2) \cdot 2M^2$ and that each of $b_i, b_i', X_1, X_2 \geq 2M^2$.

Suppose there exists a partition S_1, S_2 for P1. Then it is easy to see that P2 also has a partition T_1, T_2 such that $\sum T_1 = \sum T_2 = (2n+2)2M^2$, where $T_1 = \{b_i\}_{i|a_i \in S_1} \cup \{b_i'\}_{i|a_i \in S_2} \cup \{X_1\}$ and $T_2 = \{b_i\}_{i|a_i \in S_2} \cup \{b_i'\}_{i|a_i \in S_1} \cup \{X_2\}$.

Conversely, suppose there exists a partition for P2. This implies that there exists two subsets T_1, T_2 such that $\sum T_1 = \sum T_2 = (2n + 2)2M^2$. Now, the first thing to observe is that both X_1, X_2 cannot belong to the same set, because if so then the sum of the rest of the elements $= 4nM^2 + 2M < (2n+2)2M^2$. So let us assume that $X_1 \in T_1$ and $X_2 \in T_2$. Also, since $X_1 > b_i, \forall i$, we can remove X_1 and X_2 from the two subsets. Removing them we have $\sum T_1 - X_1 = \sum T_2 - X_2 = M + 2nM^2$. So now we can construct S_1, and S_2 as follows:

$$S_1 = \{a_i\}_{i|b_i \in T_1}$$
$$S_2 = \{a_i\}_{i|b_i \in T_2}$$

This in turn implies that P1 has a partition. □

Lemma 2 *Fixed-Difference Subset Sum is $\mathcal{NP}$-complete.*

Proof It is easy to see that Fixed-Difference Subset Sum is in $\mathcal{NP}$. To prove $\mathcal{NP}$-hardness, we show a reduction from the Partition probem. Let $\{k_1, \ldots, k_t\}$, where $\sum_i k_i = 2K$, be an arbitrary instance of Partition. Next, construct the following instance $\{\ell_1, \ldots, \ell_t, \ell_1', \ldots, \ell_t'\}$ of Fixed-Difference Subset Sum(P2), where $\ell_i = k_i + 2^{n+i}$, $\ell_i' = 2^{n+i}$, and $n = \lceil \log 2K \rceil$.

Suppose there exists a partition S_1, S_2 for P1. Then P2 has subsets T_1, T_2 such that $\sum T_1 - \sum T_2 = K + \sum_{i=1}^{t} 2^{i+n}$, where $T_1 = \{\ell_i\}_{i|k_i \in S_1} \cup \{\ell_i'\}_{i|k_i \in S_2}$ and $T_2 = \emptyset$.

Conversely, suppose there exists subsets T_1, T_2 in P2 such that $\sum T_1 - \sum T_2 = K + \sum_{i=1}^{t} 2^{i+n}$. Now, it is easy to argue that none of ℓ_i or ℓ_i' for $i = 1, \ldots, t$ can belong to T_2, because if so then the second term $(\sum_{i=1}^{t} 2^{i+n})$ of $\sum T_1 - \sum T_2$ will not be attainable. Therefore, we can construct S_1 and S_2 such that $S_1 = \{k_i\}_{i|\ell_i \in T_1}$ and $S_2 = \{k_i\}_{i|\ell_i \notin T_1}$, and this in turn implies that P1 has a partition. □

Additional proofs for Sect. 5.2

Theorem 27 *For any 3-candidate scoring rule X that is not isomorphic to plurality or veto, in single-peaked electorates, CWCM with top-truncated votes in $X_\downarrow$ is NP-complete.*

Proof (sketch) The proof here is similar to Theorem 2 and hence we only sketch the same. To prove $\mathcal{NP}$-hardness, we can proceed like in Theorem 2 by considering the following three cases. In all the cases we can use a reduction from the Partition problem.

> *Case 1 $\alpha_2 < \alpha_1 < 2\alpha_2$.* Given an arbitrary instance of Partition, construct the following instance of CWCM with $p\ L\ a\ L\ b$ as the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(p > a > b), (a > p > b), (a > b > p), (b > a > p), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S let there be a vote of weight K voting $(a > b > p)$, $(b > a > p)$, $(b > a \sim p)$, and $(p > a \sim b)$ each, and let each k_i in T have a vote of weight k_i. Now, proceed like in Theorem 2.
> *Case 2 $\alpha_1 > 2\alpha_2$.* In this case, construct the following instance of CWCM with $a\ L\ p\ L\ b$ as the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S let there be a vote of weight $(2\alpha_1 - \alpha_2)K$ voting $(a > p > b)$ and $(b > p > a)$ each, and let each k_i in T have a vote of weight $(\alpha_1 - 2\alpha_2)k_i$. Now, proceed like in Theorem 2.
> *Case 3 $\alpha_1 = 2\alpha_2$.* In this case, construct the following instance of CWCM with $a\ L\ p\ L\ b$ be the linear ordering over the candidates. In S let there be a vote of weight $3K$ voting $(a > b \sim p)$ and $(b > a \sim p)$ each, and let each k_i in T have a vote of weight k_i. Now, proceed like in Theorem 2. □

Theorem 28 *For any 3-candidate scoring rule X that is not isomorphic to plurality, in single-peaked electorates, CWCM with top-truncated votes in X_{av} is NP-complete.*

Proof (sketch) The proof here is similar to Theorem 3 and hence we only sketch the same. To prove $\mathcal{NP}$-hardness, we use a reduction from Fixed-difference Subset Sum. Given an arbitrary instance of Fixed-difference Subset Sum, construct the following instance of CWCM with $a\ L\ p\ L\ b$ as the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S let there be a vote of weight $8K$ voting $(a > b \sim p)$ and $(b > a \sim p)$ each, vote of weight $3K$ voting $(p > b > a)$, and a vote of weight K voting $(p > a > b)$. In T let each k_i have a vote of weight $2k_i$. Now, proceed like in Theorem 3. □

Lemma 3 *During each round of the eliminate(Veto), only the rightmost or the leftmost candidate in the linear ordering L is eliminated.*

Proof Let $c_1\ L\ c_2\ L\ \cdots\ L\ c_m$ be any arbitrary linear ordering. It is clear that in the first round of eliminate(Veto) only c_1 or c_m will be eliminated since all the votes only have either of them placed at the end. Also, since the votes are single-peaked, for any candidate c_j, the only candidates which can be immediately on top of it in a vote where c_j is placed last are its right and left neighbours, where the right neighbour of the rightmost candidate is the leftmost one and vice-versa. Therefore, in particular, if c_1 is the candidate eliminated in the first round then all its last votes gets transferred to either c_2 or c_m. Hence in the subsequent round only one of c_2 or c_m will be eliminated. Continuing this way we can see that at every round r only one of the corner candidates in L (which now is the linear ordering over the remaining $m - r + 1$ candidates) will be eliminated. □

Lemma 4 *The reverse of e is a single-peaked order with respect to the given linear order L.*

Proof Our claim is that $O : (p = c_m) > c_{m-1} > \cdots > c_1$ is a valid single-peaked order with respect to L. To prove this, let us assume the contrary. This implies that there exists three candidates c_i, c_j, c_k such that $c_j >_O c_i >_O c_k$, but according to the linear ordering L it is only possible that $c_i \ L \ \cdots \ L \ c_k \ L \ \cdots \ L \ c_j$ or $c_j \ L \ \cdots \ L \ c_k \ L \ \cdots \ L \ c_i$. Now, $c_j >_O c_i >_O c_k$ implies that, among c_i, c_j, and c_k, c_k is eliminated first, followed by c_i, and then c_j. But then, since at any round r only the corner elements can be eliminated (Lemma 3), c_k can never be eliminated before both c_i and c_j in either of the two linear orderings. Hence e cannot be a valid elimination order if its reverse is not single-peaked. □

6.1 Additional proofs for Sect. 5.3

Lemma 5 *For the linear order a L p L b, when the input elimination order is (a, p, b), if there exists a successful bribery, then it can be induced by the elimination order (a, b, p) (or in other words, we do not need to consider any other elimination orders).*

Proof Since the linear order is $a \ L \ p \ L \ b$, only the following votes are allowed: $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a)\}$. Let us assume that the total weight of all the voters voting $(a > p > b)$ is W_1, those voting $(p > a > b)$ is W_2, those voting $(p > b > a)$ is W_3, and those voting $(b > p > a)$ is W_4. Now, since the input elimination order is (a, p, b), we have the following inequalities: (1) $W_1 + W_2 \le W_3 + W_4$ (since a is getting eliminated in the first round, $LV(a) \ge LV(b)$, where LV(a) is the total weight of all the voters who place a at the end), and (2) $W_1 + W_2 + W_3 < W_4$ (since p gets eliminated in the second round). A successful bribery for p can proceed in two ways. First is if the bribery induces the elimination order (a, b, p) and the other way is to induce the elimination order (b, a, p). The task here is to prove that inducing (a, b, p) is better than inducing (b, a, p), or in other words, prove that inducing (b, a, p) requires bribing at least one more voter than what is required to induce (a, b, p).

To prove this, let us consider the case when the elimination order (a, b, p) is induced. Since the input already has a getting eliminated in the first round, we only need to look at the second round and see how much total weight should be bribed so that p is saved from elimination. Once a is eliminated, $LV(b) = W_1 + W_2 + W_3$, and $LV(p) = W_4$. Therefore, for p to win in the second round, the weight of votes we need to bribe is at least $D_{pb} = \left\lceil \frac{W_4 - (W_1 + W_2 + W_3)}{2} \right\rceil$. Let this bribery require k_1 voters to be bribed. Moreover, note that all the bribed votes are of the form $(b > p > a)$ and they will be bribed to vote $(p > b > a)$.

Now, let us consider the second elimination order (b, a, p). To induce this, that is to make b get eliminated in the first round, the total weight of votes we need to bribe is at least $D_{ab} = \left\lceil \frac{(W_3 + W_4) - (W_1 + W_2)}{2} \right\rceil = \left\lceil \frac{2W_3 + W_4 - (W_1 + W_2 + W_3)}{2} \right\rceil = W_3 + \left\lceil \frac{W_4 - (W_1 + W_2 + W_3)}{2} \right\rceil = W_3 + D_{pb}$. As a result, among voters voting $(b > p > a)$ we need to bribe a weight of at least D_{pb} so as to get b eliminated in the first round. But then, this bribery requires at least k_1 voters (from above), which in turn implies that we need at least one more voter to be bribed here than in the case where the elimination order (a, b, p) was induced. Hence, when the input elimination order is (a, p, b) and the linear order $aLpLb$, for a successful bribery it is enough to induce the elimination order (a, b, p). □

Theorem 37 *In single-peaked electorates, weighted-bribery with top-truncated votes is NP-complete for 3-candidate Baldwin's rule.*

Proof (sketch) The proof here is similar to that of Theorem 36 and hence we only sketch the same. To prove $\mathcal{NP}$-hardness, we show a reduction from Partition′. Given an arbitrary instance $\{a_1, \ldots, a_n\}$ of Partition′, construct an instance of weighted-bribery (C, V, p, k), where $C = \{a, b, p\}$ is the set of candidates, $a \, L \, p \, L \, b$ is the linear order of the candidates, and V is the set of the following voters.

1. For each a_i, construct a voter v_i whose weight is $2a_i$ and who votes for $(b > a \sim p)$. Let T be the set of all these voters.
2. Construct the following set of voters S: n voters of weight $2K$ each voting $(b > p > a)$, n voters of weight $2K$ each voting $(b > a \sim p)$, n voters of weight K each voting $(a > b \sim p)$, 2 voters of weight 1 each voting $(b > p > a)$, 2 voters of weight 1 each voting $(a > b \sim p)$, and 1 voter of weight 1 voting $(p > a \sim b)$.

Set the bribe limit $k = n$ and $V = S \cup T$. Now, proceed like in Theorem 36. □

References

1. Barberá, S. (2007). Indifferences and domain restrictions. *Analyse & Kritik*, *29*(2), 146–162.
2. Bartholdi, J., & Trick, M. A. (1986). Stable matching with preferences derived from a psychological model. *Operations Research Letters*, *5*(4), 165–169.
3. Bartholdi, J. J., Tovey, C. A., & Trick, M. A. (1992). How hard is it to control an election? *Mathematical and Computer Modelling*, *16*(8–9), 27–40.
4. Bartholdi, J. J, I. I. I., & Orlin, J. B. (1991). Single transferable vote resists strategic voting. *Social Choice and Welfare*, *8*(4), 341–354.
5. Bartholdi, J. J, I. I. I., Tovey, C. A., & Trick, M. A. (1989). The computational difficulty of manipulating an election. *Social Choice and Welfare*, *6*(3), 227–241.
6. Baumeister, D., Faliszewski, P., Lang, J., & Rothe, J. (2012). Campaigns for lazy voters: Truncated ballots. In *Proceedings of the 11th international conference on autonomous agents and multiagent systems (AAMAS)*, pp. 577–584.
7. Black, D. (1948). On the rationale of group decision-making. *The Journal of Political Economy*, *56*(1), 23–34.
8. Black, D., Newing, R. A., McLean, I., McMillan, A., & Monroe, B. L. (1958). *The theory of committees and elections*. Berlin: Springer.
9. Brandt, F., Brill, M., Hemaspaandra, E., & Hemaspaandra, L. A. (2015). Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. *Journal of Artificial Intelligence Research*, *53*, 439–496.
10. Briskorn, D., Erdélyi, G., & Reger, C. (2015). Bribery under partial information. In *Proceedings of the 2nd workshop on exploring beyond the worst case in computational social choice (EXPLORE)*.
11. Cantala, D. (2004). Choosing the level of a public good when agents have an outside option. *Social Choice and Welfare*, *22*(3), 491–514.
12. Coleman, T., & Teague, V. (2007). On the complexity of manipulating elections. In *Proceedings of the 13th Australasian symposium on theory of computing*, pp. 25–33.
13. Conitzer, V., Sandholm, T., & Lang, J. (2007). When are elections with few candidates hard to manipulate? *Journal of the ACM*, *54*(3), 14.
14. Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. D. (Eds.). (2016). *Handbook of computational social choice*. Cambridge: Cambridge University Press.
15. Copeland, A. H. (1951). A reasonable social welfare function. In *Seminar on applications of mathematics to social sciences*. University of Michigan
16. Dey, P., Misra, N., & Narahari, Y. (2016). Complexity of manipulation with partial information in voting. In *Proceedings of the 25th international joint conference on artificial intelligence (IJCAI)*, pp. 229–235.
17. Doignon, J. P., & Falmagne, J. C. (1994). A polynomial time algorithm for unidimensional unfolding representations. *Journal of Algorithms*, *16*(2), 218–233.
18. Duggan, J., & Schwartz, T. (2000). Strategic manipulability without resoluteness or shared beliefs: Gibbard–Satterthwaite generalized. *Social Choice and Welfare*, *17*(1), 85–93.
19. Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001). Rank aggregation methods for the web. In *Proceedings of the 10th international conference on world wide web*, pp. 613–622

20. Emerson, P. (2013). The original Borda count and partial voting. *Social Choice and Welfare*, *40*(2), 353–358.
21. Ephrati, E., & Rosenschein, J. S. (1993). Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the 13th international joint conference on artificial intelligence (IJCAI), vol. 93*, pp. 423–429.
22. Erdélyi, G., & Reger, C. (2016). Possible bribery in k-approval and k-veto under partial information. In *Proceedings of the international conference on artificial intelligence: methodology, systems, and applications (AIMSA)*, Berlin: Springer, pp. 299–309.
23. Erdélyi, G., Lackner, M., & Pfandler, A. (2013). Computational aspects of nearly single-peaked electorates. In *Proceedings of the 27th AAAI conference on artificial intelligence (AAAI)*, pp. 283–289.
24. Escoffier, B., Lang, J., & Öztürk, M. (2008). Single-peaked consistency and its complexity. In *Proceedings of the 18th European conference on artificial intelligence (ECAI)*, pp. 366–370.
25. Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. A. (2009a). How hard is bribery in elections? *Journal of Artificial Intelligence Research*, *35*, 485–532.
26. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L. A., & Rothe, J. (2009b). Llull and copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research, 35*, 275–341.
27. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L. A., & Rothe, J. (2009). A richer understanding of the complexity of election systems. In *Fundamental problems in computing*, Berlin: Springer, pp. 375–406.
28. Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. A. (2010). Using complexity to protect elections. *Communications of the ACM*, *53*(11), 74–82.
29. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L. A., & 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.
30. Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. A. (2014). The complexity of manipulative attacks in nearly single-peaked electorates. *Artificial Intelligence*, *207*, 69–99.
31. Fishburn, P. C. (1977). Condorcet social choice functions. *SIAM Journal on applied Mathematics*, *33*(3), 469–489.
32. Fitzsimmons, Z. (2015). Single-peaked consistency for weak orders is easy. In *Proceedings of the 15th conference on theoretical aspects of rationality and knowledge (TARK)*, pp. 127–140.
33. Fitzsimmons, Z., & Hemaspaandra, E. (2015). Complexity of manipulative actions when voting with ties. In *Proceedings of the 4th international conference on algorithmic decision theory (ADT)*, pp. 103–119.
34. Fitzsimmons, Z., & Hemaspaandra, E. (2016). Modeling single-peakedness for votes with ties. In *Proceedings of the 8th European starting AI researcher symposium (STAIRS)*, pp. 63–74.
35. Friedgut, E., Kalai, G., & Nisan, N. (2008). Elections can be manipulated often. In *2008 49th Annual IEEE symposium on foundations of computer science*, IEEE, pp. 243–249.
36. Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York, NY: W. H. Freeman & Co.
37. Gibbard, A. (1973). Manipulation of voting schemes: A general result. *Econometrica: Journal of the Econometric Society*, *41*(4), 587–601.
38. Hemaspaandra, E., & Hemaspaandra, L. A. (2007). Dichotomy for voting systems. *Journal of Computer and System Sciences*, *73*(1), 73–83.
39. Konczak, K., & Lang, J. (2005). Voting procedures with incomplete preferences. In *Proceedings of the IJCAI-05 multidisciplinary workshop on advances in preference handling*.
40. Lackner, M. (2014). Incomplete preferences in single-peaked electorates. In *Proceedings of the 28th AAAI conference on artificial intelligence (AAAI)*, pp. 742–748.
41. Lu, T., & Boutilier, C. (2013). Multi-winner social choice with incomplete preferences. In *Proceedings of the 23rd international joint conference on artificial intelligence (IJCAI)*, AAAI Press, pp. 263–270.
42. Narodytska, N., & Walsh, T. (2014). The computational impact of partial votes on strategic voting. In *Proceedings of the 21st European conference on artificial intelligence (ECAI)*, pp. 657–662.
43. Niemi, R. G., & Wright, J. R. (1987). Voting cycles and the structure of individual preferences. *Social Choice and Welfare*, *4*(3), 173–183.
44. Niou, E. M. (1987). A note on Nanson's rule. *Public Choice*, *54*(2), 191–193.
45. Pennock, D. M., Horvitz, E., & Giles, C. L. (2000). Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proceedings of the 17th National conference on artificial intelligence (AAAI)*, pp. 729–734.
46. Satterthwaite, M. A. (1975). Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, *10*(2), 187–217.

47. Walsh, T. (2007). Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd AAAI conference on artificial intelligence (AAAI), vol 7*, pp. 3–8.
48. Xia, L., & Conitzer, V. (2011). Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research*, *41*(2), 25–67.
49. Yang, Y. (2015). Manipulation with bounded single-peaked width: A parameterized study. In *Proceedings of the 14th international conference on autonomous agents and multiagent systems (AAMAS)*, pp. 77–85.