Parameterized Complexity of Shift Bribery in Iterative Elections

Aizhong Zhou
Department of Computer Science, Shandong University
Qingdao, China
azzhou@mail.sdu.edu.cn

Jiong Guo
Department of Computer Science, Shandong University
Qingdao, China
jguo@sdu.edu.cn

ABSTRACT

In an iterative voting system, candidates are eliminated in consecutive rounds until either the set of remaining candidates does not change or a fixed number of rounds is reached. In this paper, we consider four prominent iterative voting systems, which are all based on positional scoring rules. The Hare and Coombs systems are based on the plurality and veto rules, respectively, while the Baldwin and Nanson systems are based on the Borda rule. We study the resistance of these four systems against shift bribery. Hereby, we consider both constructive and destructive settings. It is known that all four iterative voting systems are resistant to shift bribery, that is, both constructive and destructive shift bribery problems are NP-hard for these voting systems. We complement these NP-hardness results by examining parameterized complexity of the shift bribery problems with respect to some natural parameters. Our results provide further evidence for the observation that shift bribery problems for iterative voting systems are computationally harder than for the corresponding non-iterative cases. In addition, our reductions apply several techniques which might be useful for proving hardness results for other iterative voting systems.

KEYWORDS

Iterative elections, parameterized complexity, shift bribery

ACM Reference Format:
Aizhong Zhou and Jiong Guo. 2020. Parameterized Complexity of Shift Bribery in Iterative Elections. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

The problem of aggregating the preferences of different agents (or voters) occurs in diverse situations and plays a fundamental role in artificial intelligence and social choice [5, 25]. Furthermore, studying the complexity of manipulative attacks on voting systems is one of the main themes in computational social choice. Besides manipulation (also referred to as strategic voting) and electoral control, bribery attacks aim at influencing the outcome of the election by bribing some voters to change their votes. Each voter is associated with a price. The total price of the voters to be bribed should not exceed a given budget. In constructive (or destructive) bribery, the briber's target is to make a specific candidate win (or lose) the election. The study of computational behavior of bribery was initiated by Faliszewski et al. [15]. For a comprehensive overview on computational results of control, manipulation, and bribery, we refer to the book chapters by Conitzer and Walsh [6] for manipulation, by Faliszewski and Rothe [17] for control and bribery, and by Baumeister and Rothe [3] for all three attacks.

Faliszewski [14] proposed a new notion of bribery, called nonuniform bribery. Under nonuniform bribery, a voter's price depends on the nature of changes. A similar notion called microbribery was considered by Faliszewski et al. [16], where the briber may choose which voter to bribe on which issue, in order to influence the outcome of the voting according to the evaluation criterion used. Swap bribery introduced by Elkind et al. [13] is a specialization of microbribery. In swap bribery, the briber asks a voter to perform a sequence of swaps in her vote and each swap changes the order of two consecutive candidates in this voter's vote. The briber pays for each swap a prespecified cost, which is one for the so-called unit price function. We study a special case of swap bribery, called shift bribery, where only the swaps involving the specific candidate are allowed. Shift bribery was introduced in [11, 13], and since then, a number of results have been achieved.

Both constructive and destructive shift bribery are polynomial-time solvable for the plurality and veto rules [13, 18]. Kaczmarczyk and Faliszewski [18] showed that destructive shift bribery is polynomial-time solvable for the Maximin and Borda rules, which contrasts with the results that the constructive shift bribery is NP-hard for Maximin [13] and Borda [12]. Motivated by the hardness results, Elkind et al. [13] provided a 2-approximation algorithm for constructive shift bribery for Borda. Elkind and Faliszewski [11] obtained approximations for Copeland, Maximin, and all positional scoring rules. In terms of parameterized complexity, Bredereck et al. [4] achieved a collection of results for constructive shift bribery. Among others, they proved that with respect to the number of affected votes, constructive shift bribery is $W[2]$-hard for Borda, Maximin, and Copeland. With the total number of swap operations, that is, the bribery budget in the case of unit price function, as parameter, the problem is fixed-parameter tractable (FPT) for Borda and Maximin, and is $W[1]$-hard for Copeland.

In this paper, we study shift bribery for four iterative voting systems. Iterative voting systems eliminate candidates in consecutive rounds until either the set of candidates does not change or a specific number of rounds is reached. We investigate four prominent voting rules applied to iterative voting systems, namely, Hare, Coombs, Baldwin, and Nanson. In each round, the Hare rule [24] eliminates the candidates with the least plurality score, while the Coombs rule [21] eliminates the candidates with the least veto score. The Baldwin rule [1] eliminates the candidates with the least Borda score, while the Nanson rule [23] eliminates the candidates, whose Borda scores are less than the current average Borda score. With all four rules, the candidate(s) eliminated in the last round represents the winner(s) of voting. Among the four voting rules, the Hare

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

rule and its variants are most widely used, for example, in Australia, India, Ireland, New Zealand, Pakistan, the UK, and the USA. Davies et al. [8] studied the complexity of manipulation problems for Nanson and Baldwin voting rules. They proved that manipulating Baldwin and Nanson voting systems is computationally more difficult than manipulating Borda voting, since it is NP-hard for a single manipulator to compute a manipulation strategy for the Baldwin and Nanson rules, while Borda manipulation is trivial for one manipulator. Maushagen et al. [22] initiated complexity study of shift bribery for Hare, Coombs, Baldwin, and Nanson rules. They achieved NP-hardness of both constructive and destructive cases for all four iterative voting systems.

From these classical complexity results, one can observe that compared to the non-iterative voting scenario with plurality or veto rules, the Hare and Coombs voting systems seem computationally more difficult to attack for shift bribery [2, 9]. However, the comparison of Borda and Baldwin/Nanson provides no such obvious gap concerning classical complexity status. Both iterative and non-iterative voting systems are resistant to constructive shift bribery [12, 22].

Motivated by these results, we examine parameterized complexity of the shift bribery problems for the four iterative voting systems with respect to some natural parameters such as the number of candidates or the number of votes. Our results provide further evidences for the observation that strategy attack for iterative voting systems is computationally harder than for the corresponding non-iterative cases. It is known that constructive and destructive shift bribery problems are trivial for both plurality and veto voting. Maushagen et al. [22] proved both problems become NP-hard in Hare and Coombs voting systems. We strengthen the result by showing that even with a small number of shift operations allowed, it is unlikely to have an efficient algorithm computing an optimal shift bribery strategy for Hare and Coombs. Only in the cases of few votes or candidates, Hare and Coombs voting systems might be vulnerable, that is, there exist FPT algorithms. Concerning the comparison between Borda and Baldwin/Nanson, we can now observe a parameterized complexity difference for computing optimal shift bribery strategy. Bredereck et al. [4] proved that constructive Borda shift bribery is fixed-parameter tractable with the number of allowed swap operations as parameter. In contrast, we show W[1]-hardness for this parameterization in Baldwin/Nanson systems. Further, we achieve W[1]-hardness with respect to the number of votes and FPT results with respect to the number of candidates for Baldwin and Nanson. Table 1 gives an overview of our results.

2 PRELIMINARIES

2.1 Election and voting system

An election is specified as a pair (C, V) with $C = \{c_1, \ldots, c_m\}$ being a set of candidates and $V = \{v_1, \ldots, v_n\}$ a profile of the voters' preferences over C, typically given by a multiset of linear orders of the candidates, also called votes. For example, given $C = \{c_1, c_2, c_3, c_4\}$, a vote $c_1 > c_2 > c_3 > c_4$ means c_1 is most preferred and c_4 is least preferred for a voter. A voting rule is a function that maps each election to a subset W of C, where the candidate(s) in W is(are) the winner(s) of the election. Positional scoring rules form an important class of voting systems. A positional scoring rule is defined as a scoring vector $\alpha =< \alpha_1, \ldots, \alpha_m >$, where $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$. This means that the candidate ranked at the i-th position of a vote v receives α_i points from v. The candidate receiving the most points from all votes wins the election. The most prominent scoring rules are plurality, veto, and Borda.

- In plurality, each vote gives the top-ranked candidate one point, $\alpha =< 1, 0, 0, \cdots, 0 >$;
- in veto, each vote gives all except the bottom-ranked candidate one point, $\alpha =< 1, 1, \cdots, 1, 0 >$;
- in Borda, each vote gives the candidate in the i-th position $m - i$ points, $\alpha =< m - 1, m - 2, m - 3, \cdots, 0 >$.

The score of a candidate is the total points given by all votes.

2.2 Iterative voting system

In an iterative voting system with scoring rules, the winner is determined in consecutive rounds. In each round, the candidates with the lowest score (or scores lower than the current average score) are eliminated. If in a round, all candidates have the same score (there may be only one candidate), then these candidates are the winners. The voting systems of Hare, Coombs, and Baldwin use plurality, veto, and Borda scores, respectively, and each round eliminates the candidates with the least score. In the Nanson system, each round eliminates all candidates who have Borda scores less than the current average score.

2.3 Shift Bribery

We now define the ε-Shift-Bribery problem where $\varepsilon \in$ {Hare, Coombs, Baldwin, Nanson}. In shift bribery problems, only swaps involving the specific candidate are allowed. In this paper, we consider only the unit price function, that is, each swap operation has a cost of one.

> ε**-Constructive-Shift-Bribery** (ε-CSB)
> **Input**: An election (C, V) with n votes and m candidates, a specific candidate $p \in C$, a budget $B \in \mathbf{N}$.
> **Question**: Is it possible to make p the unique winner of the election according to the ε voting system by performing at most B swap operations?

The defination of ε-DSB is similar to ε-CSB. The aim of ε-CSB is to make a specific candidate p to be the unique winner while ε-DSB is to prevent p from being the unique winner. In this paper, we focus on the unique winner model; the case seeks for making p a co-winner can be processed in a similar way. We study the parameterized complexity with respect to three parameters: m the number of candidates, n the number of votes, and B the number of swap operations.

Throughout this paper, we use $\overrightarrow{X}$ to denote an arbitrary but fixed ordering of the elements in X and $\overleftarrow{X}$ the reversed ordering of $\overrightarrow{X}$.

2.4 Parameterized complexity

Parameterized complexity allows to give a more refined analysis of computational problems and can provide a deep exploration of the connection between the problem complexity and various problem specific parameters in particular. The main hierarchy of parameterized complexity classes is: FPT $\subseteq$ W[1] $\subseteq$ W[2] ... $\subseteq$ W[t] ... $\subseteq$ XP.

Table 1: Parameterized complexity of constructive and destructive shift bribery for Hare, Coombs, Baldwin, and Nanson systems. m **: number of candidates,** n **: number of votes,** B **: number of swap operations.**

	Hare		Coombs		Baldwin		Nanson	
	Constructive	Destructive	Constructive	Destructive	Constructive	Destructive	Constructive	Destructive
B	**W[1]-h** [Thm.3.2]	**W[1]-h** [Thm.3.2]	**W[1]-h** [Thm.3.2]	**W[1]-h** [Thm.3.2]	**W[1]-h** [Thm.3.3]	**W[1]-h** [Thm.3.4]	**W[1]-h** [Thm.3.3]	**W[1]-h** [Thm.3.4]
m	**FPT** [Thm. 3.1]	**FPT** [Thm. 3.1]	**FPT** [Thm. 3.1]	**FPT** [Thm. 3.1]	**FPT** [Thm. 3.1]	**FPT** [Thm. 3.1]	**FPT** [Thm. 3.1]	**FPT** [Thm. 3.1]
n	**FPT** [Thm.3.5]	**FPT** [Thm.3.5]	OPEN	OPEN	**W[1]-h** [Thm.3.6]	OPEN	**W[1]-h** [Thm.3.6]	**W[1]-h** [Thm.3.7]

A problem is FPT (stands for "fixed-parameter tractable"), if it admits $O(f(k)\cdot|\mathcal{I}|^{O(1)})$-time algorithm, where $\mathcal{I}$ is the input, k is the parameter, and f can be any computable function. The class W[1] is the basic fixed-parameter intractability class. For more details on parameterized complexity we refer to [7, 10].

3 OUR RESULTS

3.1 The number of candidates

We first consider the parameterization with respect to the number of candidates m. Both constructive and destructive cases are FPT with respect to m for all four voting systems.

Theorem 3.1. *ε-CSB and ε-DSB are FPT with respect to the number of candidates m, where $\varepsilon \in$ {Hare, Coombs, Baldwin, Nanson}.*

Proof. Let $V = \{v_1, \cdots, v_n\}$ and $C = \{c_1, \cdots, c_m\}$ denote the vote and candidate sets, respectively. The algorithm enumerates all possible elimination orders of candidates and for each elimination order, decides by solving an integer linear program (ILP) whether it is possible to transform the given votes by at most B swap operations to a set of votes where the candidates can be eliminated according to the order by the respective voting rules. An elimination order specifies the order of candidate eliminations. For example, given $C = \{c_1, c_2, c_3, c_4\}$, $(\{c_1, c_2\}, \{c_3\}, \{c_4\})$ represents an elimination order, meaning that candidates c_1 and c_2 are eliminated in the first round, c_3 in the second round, and c_4 in the third round. Clearly, there are at most $2^m \cdot m!$ many different elimination orders.

Next, we give the ILP formulation. Note that given m candidates, there are at most $m!$ many different vote types. We say that two votes in V are of the same type if they have the same preference. Let H be the set of all vote types. Clearly, $|H| \leq m!$. For each vote type $h \in H$, let n_h be the number of votes in V of type h. We define now the variables of ILP. For each pair of vote types h and h', we define an integer variable $x_{h,h'}$, where $x_{h,h'} \geq 0$ means that $x_{h,h'}$ many votes in V of type h can be transformed to votes of type h' by performing swap operations. Given two types h and h', the number of swap operations needed to transform a vote of type h to a vote of type h' is clearly fixed and computable in polynomial time, denoted as $S_{h,h'}$. Note that some transformations from one vote type h to another type h' are not possible by swapping p with other candidates; in this case, we set $S_{h,h'} = B + 1$ with B being the number of allowed swap operations.

Given an elimination order e, let r_e be the number of elimination rounds and C_i with $0 \leq i \leq r_e$ be the set of candidates eliminated in the i-th round. In the above example, $r_e = 3, C_0 = \emptyset, C_1 = \{c_1, c_2\}, C_2 = \{c_3\}$, and $C_3 = \{c_4\}$. We set $C'_i = \bigcup_{j=0}^{i} C_j$ and $\overline{C'_i} = C \setminus C'_i$ for $0 \leq i \leq r_e$. Clearly, $C = C'_{r_e} = \bigcup_{j=0}^{r_e} C_j = \overline{C'_0}$ and $\overline{C'_{r_e}} = \emptyset$. For a subset $X \subseteq C$, let $V[X]$ be the multiset of votes, which are constructed by removing all candidates in $C \setminus X$ from all votes in V. Let $\text{score}(c_i, X, h)$ denote the points that c_i receives from a vote in $V[X]$, which is constructed from a vote in V of type h. The ILP instance for an elimination order e consists of the following constraints which can also be derived from the framework in [19]:

1. $\sum_{h' \in H} x_{h,h'} = n_h, \forall\, h \in H;$
2. $\sum_{h \in H} \sum_{h' \in H} (x_{h,h'} \cdot S_{h,h'}) \leq B;$
3. $\sum_{h' \in H} \sum_{h \in H} (x_{h,h'} \cdot \texttt{score}(c, \overline{C'_i}, h')) = \sum_{h' \in H} \sum_{h \in H} (x_{h,h'} \cdot \texttt{score}(c', \overline{C'_i}, h')),$
   $\forall\, 0 \leq i \leq r_e - 1, \forall\, c, c' \in C_{i+1}$ with $|C_{i+1}| \geq 2;$
4. $\sum_{h' \in H} \sum_{h \in H} (x_{h,h'} \cdot \texttt{score}(c, \overline{C'_i}, h')) < \sum_{h' \in H} \sum_{h \in H} (x_{h,h'} \cdot \texttt{score}(c', \overline{C'_i}, h')),$
   $\forall\, 0 \leq i \leq r_e - 1, \forall\, c \in C_{i+1}, \forall\, c' \in \overline{C'_{i+1}};$
5. $x_{h,h'} \in \{0, 1, 2, ..., n\}, \forall\, h, h' \in H.$

The first equality guarantees that for each $h \in H$, the number of votes of type h, which are transformed to other types, is equal to the number of votes of this type in V. Note that h' could be h, meaning that $x_{h,h}$ many votes of type h remain unchanged. This equality also makes sure that after the transformation there are exactly n votes. The second equality means that the total number of swap operations needed for the transformation does not exceed the budget B. The third equality and the fourth inequality guarantee that the votes after the transformation admit the same elimination order as e. More precisely, the third equality means that the candidates, who are eliminated in the $(i + 1)$-th round, should have the same score after the i-th round. The condition that the candidates, who are eliminated in the $(i+1)$-th round, have a score less than the scores of the candidates, who remain after the $(i+1)$-th round, is guaranteed by the fourth inequality. Therefore, the solution of the ILP instance gives a transformation of V, resulting in n votes, which eliminate the candidates according to the order e.

The algorithm works for both constructive and destructive shift bribery. The only difference lies in the elimination orders to be examined by ILP: the constructive case considers only the orders, where only the specific candidate p is eliminated in the last round, whereas the destructive case examines all other orders. Note that the enumeration of elimination orders and the ILP formulation are the same for Hare, Coombs, and Baldwin rules. Only score(c, X, h) is calculated according to respective definitions of scores. For the Nanson rule, we need to replace the third and forth (in)equalities by the following inequality.

$$\sum_{h' \in H} \sum_{h \in H} (x_{h,h'} \cdot \mathsf{score}(c, \overline{C'_i}, h')) \cdot |\overline{C'_i}|$$
$$< \sum_{c' \in \overline{C'_i}} \sum_{h' \in H} \sum_{h \in H} (x_{h,h'} \cdot \mathsf{score}(c', \overline{C'_i}, h'))$$
$$\forall\ 0 \leq i \leq r_e - 1, \forall\ c \in C_{i+1}.$$

Since the number of the variables of the ILP instance is bounded by $(m!)^2$, it is solvable in FPT time with m as parameter [20]. The theorem follows then from the number $m! \cdot 2^m$ of elimination orders. □

3.2 The number of swap operations

In the following, we consider the case with the number B of swap operations as parameter. We show $W[1]$-hard results of both constructive and destructive cases for all four iterative voting systems. We consider first Hare and Coombs.

Theorem 3.2. *Hare-CSB, Hare-DSB, Coombs-CSB, and Coombs-DSB are $W[1]$-hard with respect to the parameter B.*

Proof. We prove the theorem for Hare-CSB by giving a reduction from **Independent Set**. The hardness results of other problems can be proven by similar reductions. An independent set I in an undirected graph $\mathcal{G}$ is a set of pairwisely non-adjacent vertices. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and an integer k, **Independent Set** asks for a size-k independent set and is $W[1]$-hard with respect to k [10]. Let $\mathcal{V} = \{v_1, \cdots, v_{n'}\}$ and $\mathcal{E} = \{e_1, \cdots, e_{m'}\}$. Without loss of generality, assume $k \geq 2, n' > 2k + 1$ and $m' \geq n'$. We use $\mathsf{deg}(v)$ to denote the degree of v in $\mathcal{G}$. We construct a Hare-CSB instance (C, V, B) from $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), k)$ as follows.

For each vertex $v_i \in \mathcal{V}$, we create a vertex candidate $c_i \in C_1$, and for each edge $e_\ell \in \mathcal{E}$, we create an edge candidate $d_\ell \in C_2$. We also create two dummy candidate sets C_3 and C_4 with $|C_3| = |C_4| = k$. Let $C := C_1 \cup C_2 \cup C_3 \cup C_4 \cup \{p\}$ and $B := k$. The vote set V consists of five subsets $V := V_1 \cup V_2 \cup V_3 \cup V_4 \cup V_5$, which are constructed as follows. Hereby, if we do not give an explicit order for the candidates in a set, then the candidates in the set can be ordered arbitrarily.

- For each vertex $v_i \in \mathcal{V}$, we create one vote v_i^1 in V_1:
  $v_i^1 : c_i > p > C_3 > C_4 > (C_1 \setminus \{c_i\}) \cup C_2$,
  and $n' - \mathsf{deg}(v_i) - 1$ identical votes v_i^j in V_2:
  $v_i^j : c_i > (C_1 \setminus \{c_i\}) > C_2 > C_3 > C_4 > p$ for $2 \leq j \leq n' - \mathsf{deg}(v_i)$.
- For each edge $e_\ell = \{v_i, v_j\} \in \mathcal{E}$, we create two votes in V_3:
  $v_\ell^1 : c_i > c_j > d_\ell > (C_1 \setminus \{c_i, c_j\}) > (C_2 \setminus \{d_\ell\}) > C_3 > C_4 > p$,
  $v_\ell^2 : c_j > c_i > d_\ell > (C_1 \setminus \{c_i, c_j\}) > (C_2 \setminus \{d_\ell\}) > C_3 > C_4 > p$;
  and n' identical votes in V_4:
  $v_\ell^r : d_\ell > (C_2 \setminus \{d_\ell\}) > C_3 > p > C_4 > C_1$ for $3 \leq r \leq n' + 2$.
- In addition, we create $n' + 1 - k$ identical votes in V_5:
  $v^j : p > C_3 > C_4 > C_1 > C_2$ for $1 \leq j \leq n' + 1 - k$.

The current plurality scores of the candidates are: $\mathsf{score}(p) = n' + 1 - k$, $\mathsf{score}(c) = n'$ for $c \in C_1 \cup C_2$, $\mathsf{score}(c) = 0$ for $c \in C_3 \cup C_4$. Thus, the dummy candidates in $C_3 \cup C_4$ are eliminated in the first round. Note that removing the candidates in $C_3 \cup C_4$ from the votes does not change the plurality scores of the remaining candidates. Thus, the candidate p is eliminated in the second round. The candidate p is not the unique winner. Now, we show the equivalence between the **Independent Set** instance and the instance of Hare-CSB.

"$\Longrightarrow$": Suppose that there exists a size-k independent set $I = \{v_{i_j} : 1 \leq j \leq k\}$. We swap p with each vertex candidates c_{i_j} in the corresponding vote $v_{i_j}^1$ of V_1. Clearly, we need $k = B$ swap operations, satisfying the budget restriction. Now, we calculate the plurality scores of the candidates: $\mathsf{score}(p) = n' + 1$, $\mathsf{score}(c_i) = n' - 1$ for $c_i \in C_1$ with $i \in \{i_1, \cdots, i_k\}$, $\mathsf{score}(c_i) = n'$ for $c_i \in C_1$ with $i \notin \{i_1, \cdots, i_k\}$, $\mathsf{score}(c) = n'$ for $c \in C_2$; $\mathsf{score}(c) = 0$ for $c \in C_3 \cup C_4$.

Again, the candidates in $C_3 \cup C_4$ are eliminated in the first round. Afterwards, the candidates $c_i \in C_1$ with $i \in \{i_1, \cdots, i_k\}$ are eliminated. Note that, since the corresponding vertices form an independent set, removing these k candidates increases the plurality scores of some candidates in C_1, but has no influence on the scores of the edge candidates in C_2. Thus, the next round eliminates all candidates in C_2 and some candidates in C_1. With the candidates in C_2 being removed, all votes in V_4 rank p on the top. Then, $\mathsf{score}(p) = n' \cdot m' + n' + 1$. Since in total, there are $n' \cdot m' + n'^2 - k + 1 + n'$ many votes, score$(p) > \frac{1}{2}|V|$. Moreover, since removing candidates cannot decrease the plurality score of any remaining candidate. The scores of other candidates are always less than score(p), and thus, p is the unique winner.

"$\Longleftarrow$": Suppose that we can make p the unique winner with at most B swap operations. Since $|C_3| = |C_4| = B$, we cannot swap p with the candidates in C_2 in any vote in V. Let C'_1 be the set of vertex candidates, which are swapped with p, and let $\mathcal{V}'$ be the corresponding vertex set. Let $\alpha = |C'_1|$ and β be the number of the candidates in $C_3 \cup C_4$, who are swapped with p. Note that given $|C_3| = |C_4| = B$, applying swap operations in the votes in $V_2 \cup V_3 \cup V_4$ does not change the scores of the candidates. However, swapping p with a candidate c in $C_3 \cup C_4$ in a vote in V_5 increases the score of c by at most one and decreases the score of p by at most one. Therefore, since we have $n' > 2k + 1$, the candidates in $C_3 \cup C_4$ have scores less than score(p) and thus, are first eliminated. To guarantee that p is not eliminated afterwards, it must hold $\alpha \geq B - 1$. In this way, the candidates in C'_1 are eliminated with the lowest score $n' - 1$. Then, the score of p remains $n' + 1 - B + \alpha$, and the scores of other candidates are at least n'. Since p is the unique winner, it implies $\alpha = B$. Suppose that there is an edge $e_\ell = \{v, v'\}$

with $v, v' \in \mathcal{V}'$. Then, after the candidates in C_1' are eliminated, we have $\text{score}(d_\ell) = n' + 2$ and $\text{score}(p) = n' + 1$. Then, in the next round where the candidates with a score n' are eliminated, the candidate d_ℓ is not eliminated. As a consequence, the score of p remains $n' + 1$ and thus, p is eliminated before d_ℓ, contradicting that p is the unique winner. Therefore, there is no edge $e_\ell = \{v, v'\}$ with $v, v' \in \mathcal{V}'$, and $\mathcal{V}'$ forms an independent set of size k. □

In the following, we consider the parameterized complexity of shift bribery of Baldwin and Nanson with respect to the parameter B. We first show hardness for the constructive case.

Theorem 3.3. *Baldwin-CSB and Nanson-CSB are $W[1]$-hard with respect to the parameter B.*

Proof. We prove the theorem for Baldwin-CSB by giving a reduction from **Independent Set** on D-regular graphs. The result for Nanson-CSB can be shown by a similar reduction. A D-regular graph is a graph, where all vertices have the same degree D. **Independent Set** remains W[1]-hard with respect to the size k of independent sets on D-regular graphs. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, \cdots, v_{n'}\}$ and $\mathcal{E} = \{e_1, \cdots, e_{m'}\}$. We construct a Baldwin-CSB instance (C, V, B) as follows. Recall that $\overrightarrow{X}$ denotes an arbitrary but fixed ordering of the candidates in a set X and $\overleftarrow{X}$ the reversed ordering of $\overrightarrow{X}$. For a set of candidates C and $c_1, c_2 \in C$, we construct the following *vote pair*:

$$W(c_1, c_2) = (c_1 > c_2 > \overrightarrow{C \setminus \{c_1, c_2\}}, \overleftarrow{C \setminus \{c_1, c_2\}} > c_1 > c_2).$$

Similarly, for a set of candidates C and $c_1, c_2, c_3 \in C$, we construct the following vote pair:

$$W(c_1, c_2, c_3) = (c_1 > c_2 > c_3 > \overrightarrow{C \setminus \{c_1, c_2, c_3\}}, \overleftarrow{C \setminus \{c_1, c_2, c_3\}} > c_1 > c_3 > c_2).$$

According to the Borda rule, from the two votes in $W(c_1, c_2)$, candidate c_1 gets $|C|$ points, c_2 gets $|C| - 2$ points, and each of the other candidates gets $|C| - 1$ points. For simplicity, we compare the score of each candidate with the average Borda score. Thus, from the two votes in $W(c_1, c_2)$, we say that candidate c_1 gets 1 point, c_2 gets -1 point, and all other candidates get 0 point. Moreover, we say that with respect to $W(c_1, c_2)$, candidate c_2 "gains" one point by eliminating c_1, since c_2 gets the same points as other candidates in $C \setminus \{c_1, c_2\}$ after the elimination of c_1. Similarly, c_1 "loses" one point by eliminating c_2. By a similar analysis, each of c_2 and c_3 "gains" one point from the vote pair in $W(c_1, c_2, c_3)$ by eliminating c_1 and candidate c_1 "loses" one point by eliminating c_2 or c_3.

For each vertex $v_i \in \mathcal{V}$, we construct a vertex candidate $c_i \in C_1$ $(1 \leq i \leq n')$. For each edge $e_j \in \mathcal{E}$, we construct an edge candidate $d_j \in C_2$ $(1 \leq j \leq m')$. Moreover, we construct four special candidates $\{c_s, c_h, c_t, c_\ell\}$ and two dummy candidate sets C_3 and C_4 where $|C_3| = |C_4| = B$. Let $C := C_1 \cup C_2 \cup C_3 \cup C_4 \cup \{c_s, c_h, c_t, c_\ell, p\}$ and $B := k$. We construct the set of votes as follows:

- For each vertex v_i with $1 \leq i \leq n'$, we add two votes to V_1: $c_i > p > \overrightarrow{C_3} > \overrightarrow{C_4} > \overrightarrow{C_1 \setminus \{c_i\}} > \overrightarrow{C_2} > c_s > c_h > c_t > c_\ell$, and $c_\ell > c_t > c_h > c_s > \overleftarrow{C_2} > \overleftarrow{C_1 \setminus \{c_i\}} > c_i > \overleftarrow{C_4} > \overleftarrow{C_3} > p$, and the following votes to V_2: two identical vote pairs $W(c_s, c_i)$, $D+3$ identical vote pairs $W(c_i, c_t)$, and $D+4$ identical vote pairs $W(c_\ell, c_i)$;
- For each edge $e_j = \{v_i, v_{i'}\}$ with $1 \leq j \leq m'$, we add the following votes to V_2: one vote pair $W(d_j, c_i, c_{i'})$ with c_i and $c_{i'}$ corresponding to v_i and $v_{i'}$ respectively, two identical vote pairs $W(d_j, c_h)$, three identical vote pairs $W(c_t, d_j)$, $D + 2$ identical vote pairs $W(c_\ell, d_j)$;
- Furthermore, we add the following votes to V_2: two identical vote pairs $W(c_h, c_s)$, $n' - D - B - 2$ identical vote pairs $W(p, c_\ell)$, $2n' - 2B + D + 1$ identical vote pairs $W(c_\ell, c_s)$, $(n' - B)(D + 3) - D - 3$ identical vote pairs $W(c_t, c_\ell)$, and $2m' - D - 3$ identical vote pairs $W(c_h, c_\ell)$.
- We create m'^2 groups of votes in V_3, each containing the following four votes:
  - $\overrightarrow{C \setminus \{C_3 \cup C_4 \cup \{p\}\}} > \overrightarrow{C_3} > p > \overrightarrow{C_4}$,
  - $p > \overleftarrow{C_3} > \overleftarrow{C \setminus \{C_3 \cup C_4 \cup \{p\}\}} > \overleftarrow{C_4}$,
  - $\overrightarrow{C \setminus \{C_3 \cup C_4 \cup \{p\}\}} > \overrightarrow{C_4} > p > \overrightarrow{C_3}$,
  - $p > \overleftarrow{C_4} > \overleftarrow{C \setminus \{C_3 \cup C_4 \cup \{p\}\}} > \overleftarrow{C_3}$.

Note that in all votes of V_2, candidate p is always in the middle of C_3 and C_4, that is, $\overrightarrow{C_3} > p > \overrightarrow{C_4}$ or $\overleftarrow{C_4} > p > \overleftarrow{C_3}$. Let $V := V_1 \cup V_2 \cup V_3$.

The current Borda scores of the candidates are as follows. Again, we compute the difference between the score of a candidate and the average Borda score: $\texttt{score}(p) = -D - k - 2, \texttt{score}(c_i) = -D - 2$ for $c_i \in C_1, \texttt{score}(d_j) = -D - 1$ for $d_j \in C_2, \texttt{score}(c_s) = 2k - D - 3, \texttt{score}(c_h) = -D - 1, \texttt{score}(c_t) = 3m' - k(D + 3) - 3 - D, \texttt{score}(c_\ell) = D(m' + k + 4) + 2n' + 2k + 9$. The candidate p is eliminated in the second round after the candidates in $C_3 \cup C_4$ are eliminated in the first round, and p is not the unique winner. Now, we show the equivalence between the **Independent Set** instance and the instance of Baldwin-CSB. Note that, according to the m'^2 groups of votes in V_3, the candidates in $C_3 \cup C_4$ are eliminated before other candidates, no matter where the at most B swap operations apply. Furthermore, the operations that swapping p with the candidates in $C_3 \cup C_4$ have no influence on the election results.

"$\Longrightarrow$": Suppose that there is a size-k independent set I in $\mathcal{G}$. Let C' contain the candidates in C_1 corresponding to the vertices in I. We swap the candidates in C' with p in the votes in V_1, which are created corresponding to the vertices in I. That is, we perform exactly $k = B$ swap operations in exactly k votes, one operation in each vote. After the operations, the scores of the candidates in C' and p are changed: $\texttt{score}(p) = -D-2$, $\texttt{score}(c_i) = -D-3$ for $c_i \in C'$. The candidates in $C_3 \cup C_4$ are still eliminated first and the candidates in C' are eliminated in the second round. Since the corresponding vertices of the candidates in C' are independent, the score of each edge candidate d_j is $-D - 1$ or $-D - 2$. There are still $n' - k$ candidates of C_1 remaining. The candidate c_s is eliminated in the third round with a score of $-D - 3$. The following elimination sequence is $c_h, C_2, c_t, C', c_\ell, p$ and, the candidate p is the unique winner. The scores of the candidates in each round are shown in Table 2.

"$\Longleftarrow$": Suppose that there is no size-k independent set in $\mathcal{G}$. According to the construction of votes, no matter how p is swapped with other candidates, the candidates in $C_3 \cup C_4$ are eliminated before other candidates. The operations swapping p with the candidates in $C_3 \cup C_4$ do not change the final winner. Therefore, it is

Table 2: The scores of the candidates in each round in the proof of Theorem 3.3. The set C' contains the candidates, who are swapped with p. With d_j^1 we denote the edge candidates, whose corresponding edges are incident to some vertices in I, and d_j^2 denotes the other edge candidates. The candidates in $C_3 \cup C_4$ are eliminated before C' and their scores are omitted.

	p	c_i		d_j		c_s	c_h	c_t	c_ℓ
Initial	$-D-k-2$	$-D-2$		$-D-1$		$2k-D-3$	$-D-1$	$3m'-(k+1)(D+3)$	$D(m'+k'+4)+2k'+2n'+9$
		$c_i \notin C'$	$c_i \in C'$						
After swapping	$-D-2$	$-D-2$	$-D-3$	$-D-1$		$2k-D-3$	$-D-1$	$3m'-(k+1)(D+3)$	$D(m'+k'+4)+2k'+2n'+9$
				d_j^1	d_j^2				
Eliminating C'	$-D-2$	$-D-2$	—	$-D-1$	$-D-2$	$-D-3$	$-D-1$	$3m'-3-D$	$D(m'+4)-2k'+2n'+9$
Eliminating c_s	$-D-2$	$-D$	—	$-D-1$	$-D-2$	—	$-D-3$	$3m'-3-D$	$Dm'+3D+8$
Eliminating c_h	$-D-2$	$-D$	—	$-D-3$	$-D-4$	—	—	$3m'-3-D$	$Dm'+2D+2m'+5$
Eliminating d_j	$-D-2$	0	—	—	—	—	—	$-3-D$	$2D+5$
Eliminating c_t	$-D-2$	$-D-3$	—	—	—	—	—	—	$(n'-k')(D+3)+D+2$
Eliminating c_i	$n'-k-D-2$	—	—	—	—	—	—	—	$-n'+k'+D+2$
Eliminating c_ℓ and p win	$\|V\|$								

only meaningful to swap p with the candidates of C_1 in the votes of V_1. Furthermore, if the number of swap operations is less than k, p will be eliminated after $C_3 \cup C_4$. In order to make p the unique winner, we have to swap p with exactly $k = B$ vertex candidates in V_1. Clearly, these operations happen in exactly B votes in V_1, one in each vote. Let C' denote the set of vertex candidates, who are swapped with p. Since there is no size-k independent set in $\mathcal{G}$, the elimination of the candidates in C' results in an edge candidate d_j of a score $-D-3$. Then, d_j and c_s are eliminated together in the third round, which makes c_h gain at least one point and leads to the elimination of p in the fourth round. Then, p is not the unique winner. In summary, if there is no size-k independent set in $\mathcal{G}$, then candidate p cannot be an unique winner by at most k swap operations. □

Next, we consider the destructive cases of Baldwin and Nanson.

Theorem 3.4. *Baldwin-DSB and Nanson-DSB are $W[1]$-hard with respect to the parameter B.*

Proof. We prove the theorem for Nanson-DSB by giving a reduction from **Clique** on D-regular graphs. The result for the Baldwin-DSB can be shown in a similar way. Given a D-regular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and an integer k, asking for a size-k clique is $W[1]$-hard with respect to k [10]. Let $\mathcal{V} = \{v_1, \cdots, v_{n'}\}$ and $\mathcal{E} = \{e_1, \cdots, e_{m'}\}$. We construct a Nanson-DSB instance (C, V, B) as follows. Again, we compare the score of each candidate with the average Borda score. Thus, from the two votes in $W(c_1, c_2)$ in the proof of Theorem 3.3, candidate c_1 gets 1 point, c_2 gets -1 point, and all other candidates get 0 point. Moreover, we say that with respect to $W(c_1, c_2)$, candidate c_2 gains one point by eliminating c_1, and c_1 loses one point by eliminating c_2. Similarly, each of c_2 and c_3 gains one point from $W(c_1, c_2, c_3)$ by eliminating c_1 and c_1 loses one point by eliminating c_2 or c_3.

For each vertex $v_i \in \mathcal{V}$, we create a vertex candidate $c_i \in C_1$ ($1 \le i \le n'$). For each edge $e_j \in \mathcal{E}$, we create an edge candidate $d_j \in C_2$ ($1 \le j \le m'$). Moreover, we create six special candidates $C_5 = \{c_{t_1}, c_{t_2}, c_{q_1}, c_{q_2}, c_{q_3}, c_h\}$ and two dummy candidate sets C_3 and C_4 with $|C_3| = |C_4| = B$. Let $C := C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_5 \cup \{p\}$ and $B := k$. We construct the set of votes as follows:

- For each vertex v_i with $1 \le i \le n'$, add two votes $c_i > p > \overrightarrow{C_3} > \overrightarrow{C_4} > \overrightarrow{C_1 \setminus \{c_i\}} > \overrightarrow{C_2} > \overrightarrow{C_5}$ and $\overleftarrow{C_5} > \overleftarrow{C_2} > \overleftarrow{C_1 \setminus \{c_1\}} > c_i > \overleftarrow{C_4} > \overleftarrow{C_3} > p$ to the set V_1. Add D identical vote pairs $W(c_i, c_h)$ to V_2;
- For each edge $e_j = \{v_i, v_{i'}\}$ with $1 \le j \le m'$: Add one vote pair $W(d_j, c_i, c_{i'})$ and two identical vote pairs $W(c_{t_1}, d_j)$ to V_2;
- Add the following votes to V_2: $m' + k(k-1)$ identical vote pairs $W(c_{t_2}, c_{t_1})$, m' identical vote pairs $W(c_{t_1}, p)$, and one vote pair $W(p, c_{t_2})$;
- Add the following votes to V_3: m' identical vote pairs $W(p, c_{q_1})$, m' identical vote pairs $W(c_{q_1}, c_{q_2})$, and m' identical vote pairs $W(c_{q_2}, c_{q_3})$.

Note that in all votes in $V \setminus V_1$, candidate p is always in the middle of C_3 and C_4, that is, $\overrightarrow{C_3} > p > \overrightarrow{C_4}$ or $\overleftarrow{C_4} > p > \overleftarrow{C_3}$. Let $V := V_1 \cup V_2 \cup V_3$. The candidates $c_{q_1}, c_{q_2}, c_{q_3}$ and the votes in V_3 make sure that the candidate p has a score greater than the average Borda score in the first four rounds. The equivalence between the two instances can be proved in a similar but more tricky way as in the proof in Theorem 3.3. □

3.3 The number of votes

In the following, we show FPT results for Hare-CSB and Hare-DSB, and $W[1]$-hard results for Baldwin-CSB, Nanson-CSB, and Nanson-DSB with the number of votes n as parameter.

Theorem 3.5. *Hare-CSB and Hare-DSB are FPT with respect to the number of votes n.*

Proof. Let $V = \{v_1, \cdots, v_n\}$ and $C = \{c_1, \cdots, c_m\}$ denote the vote and candidate sets, respectively. We prove only the constructive case. The destructive case can be proved in a similar way. The case of $m \le n$ follows directly from Theorem 3.1. For the case $m > n$, observe that there are at most n candidates, who can get at least one point. Other candidates are eliminated with 0 point in the first round. Therefore, we enumerate all 2^n subsets of votes, which represent the votes in a possible solution, that rank p at the top. For each subset $\{i_1, \cdots, i_k\}$ with $1 \le i_1 < \cdots < i_k \le n$, we calculate the number of swap operations needed to shift p in v_{i_j} to the first position for each $1 \leq j \leq k$. If the total number of swap operations exceeds B, we exclude this subset from further consideration; otherwise, we calculate the plurality scores of the candidates with the votes $v'_1, \cdots, v'_n$, where $v'_i = v_i$ for $i \notin \{i_1, \cdots, i_k\}$ and for $i \in \{i_1, \cdots, i_k\}$, v'_i ranks the candidates in the same orders as v_i with the possible exception that p is in the first position. Then, after eliminating the candidates with the least plurality score in the first round, there remain at most n candidates. The optimal shift strategy in the votes v_i with $i \notin \{i_1, \cdots, i_k\}$ can be then computed with the ILP approach given in the proof of Theorem 3.1. In summary, we solve at most 2^n ILP's, each solvable in FPT time. This completes the proof. □

In the following, we show the hardness results for Baldwin-CSB and Nanson-CSB.

Theorem 3.6. *Baldwin-CSB and Nanson-CSB are* $W[1]$*-hard with respect to the parameter* n.

Proof. We prove the theorem for Nanson-CSB by giving a reduction from the **Multi-colored Independent Set** problem. The result for Baldwin-CSB follows from a similar but more tricky reduction. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with each vertex being colored by one of k colors, **Multi-colored Independent Set** asks for a colorful independent set of size k. A colorful set contains no two vertices with the same color. A simple reduction from **Independent Set** shows that **Multi-colored Independent Set** is W[1]-hard with respect to k. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be our input instance. Without loss of generality, we assume that the number of vertices of each color is the same, the degree of each vertex is D, and there is no edge between vertices of the same color. Further, let $\mathcal{V}^i = \{v^i_1, \dots v^i_q\}$ denote the set of vertices of color i and $\mathcal{E}^i$ be the set of edges incident to the vertices of color i. It is clear that each edge is in two $\mathcal{E}^i$'s. For each vertex v, let $\mathcal{E}(v)$ denote the set of edges that are incident to v.

We construct an instance of Nanson-CSB as follows. Let $B := k(q+(q-1)D)$. The candidate set is $C = \mathcal{V}(\mathcal{G}) \cup \mathcal{E}(\mathcal{G}) \cup \{c_t, p\} \cup D \cup D' \cup F \cup F'$, where D, D', F, and F' are sets of dummy candidates and $|D| = |D'| = |F| = |F'| = B$. For each vertex v, we define an ordering $\overrightarrow{S(v)}$ as $v > \overrightarrow{\mathcal{E}(v)}$. For each color i, we define $\overrightarrow{R(i)}$ as $\overrightarrow{D'} > \overrightarrow{\mathcal{V} \setminus \mathcal{V}^i} > \overrightarrow{\mathcal{E} \setminus \mathcal{E}^i} > t > \overrightarrow{D}$. Then $\overleftarrow{\mathcal{E}(v)}$ and $\overleftarrow{R(i)}$ denote the reversed orderings of $\overrightarrow{\mathcal{E}(v)}$ and $\overrightarrow{R(i)}$, respectively. We construct the set of votes as follows.

For each color $1 \leq i \leq k$, we create four votes in V^i:

$$x_i : \overrightarrow{S(v^i_1)} > \cdots > \overrightarrow{S(v^i_q)} > p > \overrightarrow{R(i)} > \overrightarrow{F} > \overrightarrow{F'},$$
$$x'_i : \overleftarrow{S(v^i_q)} > \cdots > \overleftarrow{S(v^i_1)} > p > \overrightarrow{R(i)} > \overrightarrow{F} > \overrightarrow{F'},$$
$$y_i : \overleftarrow{R(i)} > p > \overleftarrow{S(v^i_q)} > \cdots > \overleftarrow{S(v^i_1)} > \overleftarrow{F'} > \overleftarrow{F},$$
$$y'_i : \overleftarrow{R(i)} > p > \overleftarrow{S(v^i_1)} > \cdots > \overleftarrow{S(v^i_q)} > \overleftarrow{F'} > \overleftarrow{F}.$$

Further, we create the following six votes in V':

$$z_1 : \overrightarrow{\mathcal{E}(\mathcal{G})} > \overrightarrow{\mathcal{V}(\mathcal{G})} > \overrightarrow{F} > p > \overrightarrow{F'} > c_t > \overrightarrow{D} > \overrightarrow{D'},$$
$$z'_1 : c_t > \overleftarrow{\mathcal{V}(\mathcal{G})} > \overleftarrow{\mathcal{E}(\mathcal{G})} > \overleftarrow{F'} > p > \overleftarrow{F} > \overleftarrow{D'} > \overleftarrow{D},$$
$$z_2 : c_t > \overrightarrow{\mathcal{E}(\mathcal{G})} > \overrightarrow{\mathcal{V}(\mathcal{G})} > \overrightarrow{F} > p > \overrightarrow{F'} > \overrightarrow{D} > \overrightarrow{D'},$$
$$z'_2 : \overleftarrow{F'} > p > \overleftarrow{F} > c_t > \overleftarrow{\mathcal{V}(\mathcal{G})} > \overleftarrow{\mathcal{E}(\mathcal{G})} > \overleftarrow{D'} > \overleftarrow{D},$$
$$z_3 : p > c_t > \overrightarrow{F} > \overrightarrow{F'} > \overrightarrow{\mathcal{E}(\mathcal{G})} > \overrightarrow{\mathcal{V}(\mathcal{G})} > \overrightarrow{D} > \overrightarrow{D'},$$
$$z'_3 : \overleftarrow{\mathcal{V}(\mathcal{G})} > \overleftarrow{\mathcal{E}(\mathcal{G})} > \overleftarrow{F'} > \overleftarrow{F} > p > c_t > \overleftarrow{D'} > \overleftarrow{D}.$$

The vote set V is set equal to $(\bigcup_{i=1}^{k} V^i) \cup V'$. By the construction of the votes, we can observe that each candidate in $C \setminus (F \cup F')$ receives $|C| + 2B$ points from the votes x_i and y_i for each $1 \leq i \leq k$, while each candidate in $F \cup F'$ receives $2B - 1$ points from these two votes. In total, each candidate in $C \setminus (F \cup F')$ receives $2k(|C| + 2B)$ points from the votes in $\bigcup_{i=1}^{k} V^i$, while each candidate in $F \cup F'$ receives $2k(2B - 1)$ points. Thus, concerning the votes in $\bigcup_{i=1}^{k} V^i$, each candidate in $F \cup F'$ receives a Borda score less than the average Borda score. From the remaining six votes, we can conclude that the candidates in $D \cup D'$ have scores less than the average Borda score. It is obvious that the scores of candidates in $D \cup D' \cup F \cup F'$ are less than the average score, and thus, the first round eliminates these candidates. Afterwards, all candidates receive the same points from $\bigcup_{i=1}^{k} V^i$. However, p receives the least point from the remaining six votes. Thus, p is not the unique winner and is eliminated in the second round.

"$\Longrightarrow$": Suppose there is a colorful independent set I for $\mathcal{G}$ and for each color $1 \leq i \leq k$, let $v^i_{s_i}$ be the vertex of color i in I. For each pair of x_i and x'_i, we shift p in x_i by swap operations to the position directly in front of the candidate $v^i_{s_i+1}$ and in x'_i directly in front of the candidate $v^i_{s_i}$. For every pair of votes x_i and x'_i, $q + (q-1) \cdot D$ swaps are needed and in total B swaps are needed. Thus, in this way, all edge and vertex candidates have been swapped with p, and the score of each edge and vertex candidate is decreased by at least one, resulting in that instead of p, the vertex and edge candidates are eliminated in the second round. Then, the candidate c_t is eliminated next and p is the unique winner.

"$\Longleftarrow$": Suppose that p is the unique winner after swapping p with other candidates. Since $|D| = |D'| = |F| = |F'| = B$, p can be swapped with the dummy candidates, or the vertex and edge candidates in x_i or x'_i. No matter which candidates p is swapped with, the dummy candidates have always scores less than the average Borda score and are eliminated in the first round. Suppose that there exist vertex candidates or edge candidates, which are not swapped with p. Let C' denote the set of these candidates. Then, the candidates in $(\mathcal{V}(\mathcal{G}) \cup \mathcal{E}(\mathcal{G})) \setminus C'$ are eliminated next. If $|C'| = 1$, then all remaining candidates have the same Borda score and are eliminated together. Then p is not the unique winner. If $|C'| > 1$, the score of p is less than the average Borda score and p is eliminated after eliminating dummy candidates. Then, p is not the unique winner. To guarantee that p is the unique winner, it must satisfy $|C'| = 0$. It also means that all vertex and edge candidates have been swapped with p. For each color, p has to be swapped with at least $q+(q-1) \cdot D$ candidates and in total, exact B swaps for all colors. After these swaps, p lies directly in front of $v^i_{s_i+1}$ in x_i and directly in front of $v^i_{s_i}$ in x'_i. On the other hand, for each color i, there is a set of edge candidates $\mathcal{E}(v^i_{s_i})$, that are not swapped with p. Thus, to guarantee that the score of each edge candidate is decreased by one, there cannot be any edge between the vertices of $v^i_{s_i}$ with $1 \le i \le k$. It also means that the corresponding k vertices form an independent set. Therefore, Nanson-CSB is $W[1]$-hard with respect to the parameter n. □

Finally, we prove W[1]-hardness of the destructive case of Nanson.

THEOREM 3.7. *Nanson-DSB is $W[1]$-hard with respect to the parameter n.*

PROOF. We prove the theorem by a similar but more tricky reduction from the **Multi-colored Clique** problem. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be our input instance, i.e., an undirected graph with each vertex being colored with one of k colors. Without loss of generality, we assume that the number of vertices of each color is the same, the degree of each vertex is D, and there is no edge between vertices of the same color. Further, let $\mathcal{V}^i = \{v^i_1, \dots v^i_q\}$ denote the set of vertices of color i and $\mathcal{E}^i$ be the set of edges incident to vertices of color i. For each vertex v, let $\mathcal{E}(v)$ denote the set of edges that are incident to v.

We construct an instance of Nanson-DSB as follows. Let $B := k(q + (q + 1)D)$. The candidate set is $C := \mathcal{V}(\mathcal{G}) \cup \mathcal{E}(\mathcal{G}) \cup C_1 \cup C_2 \cup C_3 \cup \{p, c_s, c_d\} \cup H$, where H is a set of dummy candidates with $|H| = 4B$, $C_1 = \{c_{t_1}, c_{t_2}, c_{t_3}, c_{t_4}\}$, $C_2 = \{c_{r_1}, c_{r_2}, c_{r_3}, c_{r_4}\}$, and $C_3 = \{c_{u_1}, c_{u_2}, c_{u_3}, c_{u_4}, c_{u_5}\}$. Again, for each vertex v, we define $\overrightarrow{S(v)}$ as $v > \overrightarrow{\mathcal{E}(v)}$. For each color i, we define $\overrightarrow{R(i)}$ as $\overrightarrow{\mathcal{V} \setminus \mathcal{V}^i} > \overrightarrow{\mathcal{E} \setminus \mathcal{E}^i} > \overrightarrow{C_1} > \overrightarrow{C_2} > \overrightarrow{C_3} > c_s > c_d$. We construct the set of votes as follows. The set H plays the same role as the dummy candidates in the proof of Theorem 3.6. To simplify the presentation, we omit these candidates in the votes.

For each color $1 \le i \le k$, we create four votes in V^i:

$$\overrightarrow{x_i} : \overrightarrow{R(i)} > p > \overrightarrow{S(v^i_1)} > \dots > \overrightarrow{S(v^i_q)},$$
$$\overrightarrow{x'_i} : \overrightarrow{R(i)} > p > \overleftarrow{S(v^i_q)} > \dots > \overleftarrow{S(v^i_1)},$$

and $\overrightarrow{y_i} := \overleftarrow{x_i}, \overrightarrow{y'_i} := \overleftarrow{x'_i}$ ($\overleftarrow{x_i}$ and $\overleftarrow{x'_i}$ denote the reversed orderings of $\overrightarrow{x_i}$ and $\overrightarrow{x'_i}$, respectively).

We create the following seven vote pairs in V_2:

$$z_1 : \overrightarrow{C_1} > \overrightarrow{\mathcal{E}(\mathcal{G})} > \overrightarrow{C \setminus (\mathcal{E}(\mathcal{G}) \cup C_1)},$$
$$\overleftarrow{C \setminus (\mathcal{E}(\mathcal{G}) \cup C_1)} > \overleftarrow{C_1} > \overleftarrow{\mathcal{E}(\mathcal{G})};$$
$$z_2 : \overrightarrow{\mathcal{V}(\mathcal{G})} > \overrightarrow{C_1} > \overrightarrow{C \setminus (\mathcal{V}(\mathcal{G}) \cup C_1)},$$
$$\overleftarrow{C \setminus (\mathcal{V}(\mathcal{G}) \cup C_1)} > \overleftarrow{\mathcal{V}(\mathcal{G})} > \overleftarrow{C_1};$$
$$z_3 : \overrightarrow{C_2} > \overrightarrow{\mathcal{V}(\mathcal{G})} > \overrightarrow{C \setminus (\mathcal{V}(\mathcal{G}) \cup C_2)},$$
$$\overleftarrow{C \setminus (\mathcal{V}(\mathcal{G}) \cup C_2)} > \overleftarrow{C_2} > \overleftarrow{\mathcal{V}(\mathcal{G})};$$
$$z_4 : p > \overrightarrow{C_2} > \overrightarrow{C \setminus (\{p\} \cup C_2)}, \overleftarrow{C \setminus (\{p\} \cup C_2)} > p > \overleftarrow{C_2};$$
$$z_5 : \overrightarrow{C_1} > p > \overrightarrow{C \setminus (\{p\} \cup C_1)}, \overleftarrow{C \setminus (\{p\} \cup C_1)} > \overleftarrow{C_1} > p;$$
$$z_6 : \overrightarrow{\mathcal{E}(\mathcal{G})} > c_s > \overrightarrow{C \setminus (\mathcal{E}(\mathcal{G}) \cup \{c_s\})},$$
$$\overleftarrow{C \setminus (\mathcal{E}(\mathcal{G}) \cup \{c_s\})} > \overleftarrow{\mathcal{E}(\mathcal{G})} > c_s;$$
$$z_7 : p > c_s > \overrightarrow{C \setminus \{p, c_s\}}, \overleftarrow{C \setminus \{p, c_s\}} > p > c_s.$$

Further, V_3 contains the following seven vote pairs:

$$z_8 : p > c_d > \overrightarrow{\mathcal{E}(\mathcal{G})} > \overrightarrow{C \setminus \mathcal{E}(\mathcal{G})}, \overleftarrow{C \setminus \mathcal{E}(\mathcal{G})} > p > \overleftarrow{\mathcal{E}(\mathcal{G})} > c_d;$$
$$z_9 : c_d > c_{u_1} > \overrightarrow{C \setminus \{c_d, c_{u_1}\}}, \overleftarrow{C \setminus \{c_d, c_{u_1}\}} > c_d > c_{u_1};$$
$$z_{10} : c_{u_4} > c_{u_2} > \overrightarrow{C \setminus \{c_{u_2}, c_{u_4}\}}, \overleftarrow{C \setminus \{c_{u_2}, c_{u_4}\}} > c_{u_4} > c_{u_2};$$
$$z_{11} : c_{u_2} > p > \overrightarrow{C \setminus \{c_{u_2}, p\}}, \overleftarrow{C \setminus \{c_{u_2}, p\}} > c_{u_2} > p;$$
$$z_{12} : c_{u_2} > c_{u_3} > \overrightarrow{C \setminus \{c_{u_2}, c_{u_3}\}}, \overleftarrow{C \setminus \{c_{u_2}, c_{u_3}\}} > c_{u_2} > c_{u_3};$$
$$z_{13} : p > c_{u_4} > \overrightarrow{C \setminus \{p, c_{u_4}\}}, \overleftarrow{C \setminus \{p, c_{u_4}\}} > p > c_{u_4};$$
$$z_{14} : c_{u_3} > c_{u_5} > \overrightarrow{C \setminus \{c_{u_3}, c_{u_5}\}}, \overleftarrow{C \setminus \{c_{u_3}, c_{u_5}\}} > c_{u_3} > c_{u_5}.$$

Note that in V_3, there are four identical copies of z_8 and z_9, three identical copies of z_{10}, two identical copies of z_{11}; and one copy for each of other votes. Let $V_1 = \bigcup V^i$ and $V := V_1 \cup V_2 \cup V_3$ and $|V| = 4k + 46$. It is easy to verify that p is the unique winner, as $\mathcal{E}(\mathcal{G}) \cup \{c_s, c_{u_1}, c_{u_5}\}$ are eliminated in the first round, $C_1 \cup \{c_d, c_{u_3}\}$ in the second round, and $\mathcal{V}(\mathcal{G}) \cup C_2 \cup \{c_{u_2}\}$ in the third round. The role of $C_1 \cup C_2 \cup \{c_s\}$ is to control in which round $\mathcal{E}(\mathcal{G}) \cup \mathcal{V}(\mathcal{G})$ are eliminated, and $C_3 \cup \{c_d\}$ is to control in which round p is eliminated. The proof of the equivalence between the two instances is deferred to the long version. □

4 CONCLUSION

We achieved FPT and $W[1]$-hard results for both constructive and destructive shift bribery problems on the iterative voting systems of Hare, Coombs, Baldwin, and Nanson. There remain some open problems. For instance, the parameterized complexity of Baldwin-DSB, Coombs-CSB, and Coombs-DSB are open with respect to the number of votes. Moreover, we only considered the shift bribery with the unit price function. It would be interesting to study other price functions such as the all-or-nothing price function. Some of our results hold for other price functions (all FPT results), but some do not. One might think that the shift bribery problems with all-or-nothing price function could be easier to solve than the ones with unit price function, because with the all-or-nothing function, it seems reasonable to shift p to the first position in the constructive case and to the last position in the destructive case. However, there exist concrete examples, where the optimal shift strategy is to leave p in the middle of some votes. Another direction for future work can be the approximability of shift bribery problems for these systems. Furthermore, the shift bribery behavior of other iterative voting systems such as Plurality with Runoff could be an interesting research topic. Finally, we are not aware of any computational complexity result for controlling iterative voting systems.

ACKNOWLEDGMENTS

We thank the AAMAS-20 reviewers for their constructive comments. Both authors are supported by the National Natural Science Foundation of China (Grants No.61772314, 61761136017).

REFERENCES

[1] Joseph Baldwin. 1926. The technique of the Nanson preferential majority system of election. *Proceedings of the Royal Society of Victoria* (1926), 42–52.

[2] John Bartholdi and James Orlin. 1991. Single transferable vote resists strategic voting. *Social Choice and Welfare* (1991), 341–354.

[3] Dorothea Baumeister and Jörg Rothe. 2016. Preference aggregation by voting. In *Economics and Computation*. 197–325.

[4] Robert Bredereck, Jiehua Chen, Piotr Faliszewski, André Nichterlein, and Rolf Niedermeier. 2016. Prices matter for the parameterized complexity of shift bribery. *Information and Computation* (2016), 140–164.

[5] Vincent Conitzer. 2010. Making decisions based on the preferences of multiple agents. *Commun. ACM* (2010), 84–94.

[6] Vincent Conitzer and Toby Walsh. 2016. Barriers to Manipulation in Voting. In *Handbook of Computational Social Choice*. 127–145.

[7] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer.

[8] Jessica Davies, George Katsirelos, Nina Narodytska, Toby Walsh, and Lirong Xia. 2014. Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules. *Artificial Intelligence* (2014), 20–42.

[9] Jessica Davies, Nina Narodytska, and Toby Walsh. 2012. Eliminating the weakest link: Making manipulation intractable?. In *AAAI*. 1333–1339.

[10] Rodney Downey and Michael Fellows. 2012. *Parameterized Complexity*. Springer Science and Business Media.

[11] Edith Elkind and Piotr Faliszewski. 2010. Approximation algorithms for campaign management. In *WINE*. 473–482.

[12] Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. 2017. Properties of multiwinner voting rules. *Social Choice and Welfare* (2017), 599–632.

[13] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. 2009. Swap bribery. In *SAGT*. 299–310.

[14] Piotr Faliszewski. 2007. Nonuniform bribery. *Computer Science* (2007), 1569–1572.

[15] Piotr Faliszewski, Edith Hemaspaandra, and Lane Hemaspaandra. 2006. The complexity of bribery in elections. In *AAAI*. 641–646.

[16] Piotr Faliszewski, Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. 2009. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research* (2009), 275–341.

[17] Piotr Faliszewski and Jörg Rothe. 2016. Control and Bribery in Voting. In *Handbook of Computational Social Choice*. 146–168.

[18] Andrzej Kaczmarczyk and Piotr Faliszewski. 2016. Algorithms for destructive shift bribery. In *AAMAS*. 305–313.

[19] Dusan Knop, Martin Koutecký, and Matthias Mnich. 2018. A Unifying Framework for Manipulation Problems. In *AAMAS*. 256–264.

[20] Hendrik W. Lenstra. 1983. Integer Programming with a Fixed Number of Variables. *Mathematics Operations Research* (1983), 538–548.

[21] Jonathan Levin and Barry Nalebuff. 1995. An introduction to vote-counting schemes. *Journal of Economic Perspectives* (1995), 3–26.

[22] Cynthia Maushagen, Marc Neveling, Jörg Rothe, and Ann-Kathrin Selker. 2018. Complexity of shift bribery in iterative elections. In *AAMAS*. 1567–1575.

[23] Edward Nanson. 1882. Methords of election. *Transactions and Proceedings of the Royal Society of Victoria* (1882), 197–240.

[24] Alan Taylor and Allison Pacelli. 2008. *Mathematics and politics: strategy, voting, power, and proof*. Springer Science and Business Media.

[25] William Zwicker. 2016. Introduction to the Theory of Voting. In *Handbook of Computational Social Choice*. 23–56.