Parameterized Complexity of Candidate Control in Elections and Related Digraph Problems [1]

Nadja Betzler [a,2,*] Johannes Uhlmann [a,3]

[a] *Institut für Informatik, Friedrich-Schiller-Universität Jena Ernst-Abbe-Platz 2, D-07743 Jena, Germany*

Abstract

There are different ways for an external agent to influence the outcome of an election. We concentrate on "control" by adding or deleting candidates. Our main focus is to investigate the parameterized complexity of various control problems for different voting systems. To this end, we introduce natural digraph problems that may be of independent interest. They help in determining the parameterized complexity of control for different voting systems including Llull, Copeland, and plurality voting. Devising several parameterized reductions, we provide an overview of the parameterized complexity of the digraph and control problems with respect to natural parameters such as adding/deleting only a bounded number of candidates or having only few voters.

Key words: Computational social choice, digraph modification problems, W[1]-/W[2]-hardness, NP-hardness, Copeland$^\alpha$ voting, plurality voting

---

* Corresponding author.
Email addresses: `nadja.betzler@uni-jena.de` (Nadja Betzler), `johannes.uhlmann@uni-jena.de` (Johannes Uhlmann).

[1] A preliminary version of this paper appeared in *Proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA'08)*, Springer LNCS 5165, pp. 43-53.

[2] Supported by the Deutsche Forschungsgemeinschaft, Emmy Noether research group PIAF (NI 369/4) and project DARE (GU 1023/1).
[3] Supported by the Deutsche Forschungsgemeinschaft project PABI (NI 369/7).

1 Introduction

Computational social choice is an important field of interdisciplinary research. Herein, the investigation of voting systems plays a prominent role. For an overview see the two recent surveys by Chevaleyre et al. [8] and Faliszewski et al. [20]. Besides obvious classical applications in political or other elections, voting systems also have applications in multi-agent systems or rank aggregation. In addition to work that focuses on the problem to determine the winner of an election for different voting systems, there is a considerable amount of work investigating how an external agent or a group of voters can influence the election in favor or disfavor of a distinguished candidate. The studied scenarios are manipulation [2,7,11,27,30], electoral control [3,15,17,18,24,27], lobbying [9,14], and bribery [16,17,18]. Recently, parameterized algorithms and hardness results have been developed for the analysis of voting problems [4,5,6,9,17,18]. In this work, we investigate the parameterized complexity of some variants of electoral control and closely related digraph problems. We start by introducing the scenario of electoral control and give an overview of the corresponding literature.

1.1 Electoral control

An election consists of a multiset of votes and a set of candidates. Throughout this work, a vote is a preference list (permutation) of all candidates. To *control* an election, an external agent, traditionally called the *chair*, can change the voting procedure to reach certain goals. For example, a typical question is how many candidates the chair has to delete to make his/her favorite candidate a winner. In general, the considered types of control are adding, deleting, or partitioning candidates or voters [3,24]. Furthermore, one distinguishes between *constructive control* (CC), that is, the chair aims at making a distinguished candidate the winner, and *destructive control* (DC), that is, the chair wants to prevent a distinguished candidate from winning [24].

The consideration of the computational complexity of control problems goes back to Bartholdi, Tovey and Trick [3]. More precisely, they defined a voting system to be *immune* against a type of control if it is never possible for the chair to change a non-winner candidate to be a winner candidate, otherwise it is *susceptible* for the considered kind of control. Unfortunately, commonly used voting systems are susceptible to some kinds of control. For example, plurality voting is even susceptible to all standard types of control. Thus, Bartholdi et al. [3] suggested computational hardness as a favorable property of voting systems if immunity is not guaranteed. Here, one classically distinguishes between *resistant*, that means, controlling the election is NP-hard, and *vulnerable*, that is, controlling the election can be accomplished in polynomial time. Note that the term "resistant" may be misleading in the sense that it does only imply hardness for a worst-case scenario. Nevertheless, it seems interesting to investigate whether there are efficient strategies for control in general.

A series of publications [3,17,18,24] provides a complete picture of the classical computational complexity for eleven basic types of control for the standard voting systems approval, plurality, Condorcet, and Copeland$^\alpha$ (defined for all rational values of α in the range of $[0, 1]$). Additionally, in a very recent work, Erdélyi et al. [15] considered control for "sincere-strategy preference-based approval voting". Regarding parameterized complexity, Faliszewski et al. [17,18,19] obtained some first results. They considered control of Copeland$^\alpha$ voting with respect to the parameters "number of candidates" and "number of votes" for constructive and destructive control in the eleven standard control scenarios. For control by adding and deleting candidates they obtained fixed-parameter tractability with respect to the parameter "number of candidates" for all considered scenarios. The parameterized complexity with respect to the parameter "number of votes" was left open.

In this work, we focus on *candidate control*, that is, either deleting or adding candidates, both in the constructive and destructive case. [4] Among all studied voting systems only plurality, Copeland$^\alpha$, and the recently introduced sincere-strategy preference-based approval voting systems are resistant to some kind of candidate control [3,15,17,18,24]. In this work, we focus on plurality and Copeland$^\alpha$ voting, which are described in the following.

Plurality voting Plurality voting is probably the most widely used voting system, for example, it is applied in political elections. Here, every voter can vote for one candidate and the candidate with the highest number of votes wins. To evaluate the effect of adding or deleting a set of candidates, in the following we work with the full preference lists of all voters. Then, formally, in *plurality voting*, for every vote the candidate that is ranked first in the preference list gets one point. The *score* of a candidate is the total number of its points. A candidate with the highest score wins. Plurality voting is resistant to constructive and destructive control by adding and by deleting candidates [3,24].

Copeland voting *Copeland$^\alpha$ voting* is defined for all rational values of α in the range of $[0, 1]$. Throughout this work, α always denotes a rational number within $[0, 1]$. It is based on pairwise comparisons between candidates: A

[4] In contrast to Faliszewski et al. [18,19] we do not include control by partitioning the set of candidates in the definition of candidate control.

candidate wins the pairwise head-to-head contest against another candidate if it is better positioned in more than half of the votes. The winner of a head-to-head contest is awarded one point and the loser receives no point. If two candidates are tied, both candidates get α points. A *Copeland*$^{\alpha}$ winner is a candidate with the highest score. Faliszewski et al. [17] devoted their paper to the two important special cases $\alpha = 0$, denoted as *Copeland*, and $\alpha = 1$, denoted as *Llull*. Nowadays, Copeland$^{\alpha}$ elections are commonly used. For example, in sport tournaments, like chess or in football leagues, the teams or players can be considered as candidates. Regarding the complexity of control, Copeland$^{\alpha}$ voting is resistant to constructive candidate control and vulnerable for destructive candidate control [17,18].

Next, we introduce some digraph problems which are closely related to candidate control in Copeland and Llull elections.

1.2 Digraph problems

A Copeland or Llull election can be depicted by a digraph where the candidates are represented as vertices and there is an arc from vertex c to vertex d if and only if the corresponding candidate c defeats the corresponding candidate d in the head-to-head contest. Obviously, the Copeland score of a candidate equals the outdegree of the corresponding vertex and, thus, a Copeland winner corresponds to a vertex with maximum outdegree. The Llull score of a candidate c can be considered as the total number of candidates minus the number of candidates that beat c in the pairwise head-to-head contest. Thus, a Llull winner corresponds to a vertex with minimum indegree. Naturally, the deletion/addition of a vertex one-to-one corresponds to the deletion/addition of a candidate in the election. These observations motivate the introduction of the following digraph problems.

> MAX-OUTDEGREE DELETION (MOD)
> *Given:* A digraph $D = (W, A)$, a distinguished vertex $w_c \in W$, and an integer $k \geq 1$.
> *Question:* Is there a subset $W' \subseteq W \setminus \{w_c\}$ of size at most k such that w_c is the only vertex that has maximum outdegree in $D[W \setminus W']$?

Analogously, given a directed graph, a distinguished vertex, and a positive integer k, MIN-INDEGREE DELETION (MID) asks for a set of at most k vertices whose removal makes the distinguished vertex to be the only vertex with minimum indegree. We say that MID correspond to constructive control by deleting candidates for Llull voting and MOD correspond to constructive control by deleting candidates for Copeland voting. The problems for adding vertices are defined as follows:

Min-Indegree Addition (MIA)
Given: A digraph $D = (W, A)$ with vertex set $W = \mathcal{C} \uplus \mathcal{N}$, a distinguished vertex $c \in \mathcal{C}$, and an integer $k \geq 1$.
Question: Is there a subset $\mathcal{N}' \subseteq \mathcal{N}$ of at most k vertices such that c is the only vertex of minimum indegree in $D[\mathcal{C} \cup \mathcal{N}']$?

Max-Outdegree Addition (MOA)
Given: A digraph $D = (W, A)$ with vertex set $W = \mathcal{C} \uplus \mathcal{N}$, a distinguished vertex $c \in \mathcal{C}$, and an integer $k \geq 1$.
Question: Is there a subset $\mathcal{N}' \subseteq \mathcal{N}$ of at most k vertices such that c is the only vertex of maximum outdegree in $D[\mathcal{C} \cup \mathcal{N}']$?

For the addition problems we have that MIA corresponds to constructive control by adding candidates for Llull voting and MOA corresponds to constructive control by adding candidates for Copeland voting.

By the above observation that the deletion/addition of a candidate one-to-one corresponds to the deletion/addition of a vertex, every instance of a control problem can be transformed to an equivalent instance of the corresponding digraph problem. More specifically, a distinguished candidate can become the only winner of a Copeland election by deleting/adding k candidates if and only if the corresponding vertex can become the only vertex with maximum outdegree by deleting/adding k vertices in the corresponding digraph. In the same way, a distinguished candidate can become the only winner of a Llull election by deleting/adding k candidates if and only if the corresponding vertex can become the only vertex with minimum indegree by deleting/adding k vertices.

1.3 Motivation

In this work, we investigate the parameterized complexity [13,21,28] of Copeland$^\alpha$ and plurality voting, two important and commonly used voting systems. From the chair's point of view, it is interesting to find efficient strategies to reach her/his goal. There are legal control scenarios as for example persuading additional players to participate in a sport competition (like chess competitions in which usually every player plays against every other player) in order to make the favorite player the winner. And, a maybe less legal but common action is to slander a candidate to get rid of him. Since it seems plausible to add or delete only a limited number of candidates, parameterized complexity analysis is meaningful in this context. In particular, the existence of parameterized algorithms for parameters that assume presumably small values in natural voting scenarios would yield a general control strategy. Note that the goal of many publications is to show that, if control is not impossible, it is at

Table 1
Parameterized complexity of Max-Outdegree Deletion (MOD) and Min-Indegree Deletion (MID). W[2]-membership is given in Theorem 6, the other results are from [1] Theorem 1, [2] Proposition 2, [3] Theorem 2, [4] Proposition 3. Clearly, it does not make sense to consider tournaments with degree constraints.

parameters	# deleted vertices k		maximum degree d		(k, d)	
problems	MOD	MID	MOD	MID	MOD	MID
general digraphs	W[2]-c[1,3]	W[2]-c[3]	NP-c, $d \geq 3$[1]	FPT[2]	FPT[4]	FPT[2]
acyclic digraphs	W[2]-c[1]	P[2]	NP-c, $d \geq 3$[1]	P[2]	FPT[4]	P[2]
tournaments	W[2]-c[3]	W[2]-c[3]	-	-	-	-

least computationally hard (often showing NP-hardness). However, as noted by Conitzer et al. [11], such hardness results lose relevance if there are efficient fixed-parameter algorithms for realistic settings.

Regarding the digraph problems, they are natural and simple and, thus, deserve being studied on their own. Indeed, it is rather surprising that they seem not to have been considered until now. In general, the study of the parameterized complexity of digraph problems is a growing field of research (see [23] for a recent survey).

1.4 Our contributions

We provide a first study of the introduced natural digraph problems that might be of independent interest and show that they are closely related to the considered control problems. In Section 3, we investigate the computational complexity of MOD and MID (as well as MIA and MOA) for several special graph classes and parameters, providing a differentiated picture of their parameterized complexity including algorithms and intractability results (see Table 1). The main technical achievement of this section is to show that MOD and MID are W[2]-complete in tournaments. One interesting observation is that, although MOD and MID seem to be very similar, their (parameterized) complexity varies for different graph classes for several parameterizations (Table 1). Some of the considered special cases and parameterizations of the digraph problems map to realistic voting scenarios with presumably small parameters. Based on these connections and by giving new parameterized reductions, in Section 4, we provide an overview of parameterized hardness results for control problems with respect to the "number of deleted/added candidates" (Table 3). Surprisingly, for plurality voting, which can be considered as the "easiest" voting system in terms of winner evaluation and for which the Manipulation problem can be solved optimally by a simple greedy strategy [11], all kinds of candidate control are intractable from this parameterized point of view. The reductions used for the digraph problems often rely on similar ideas. In contrast, the parameterized reductions used for plurality voting require new approaches. Regarding the structural parameter "number of votes", we answer an open question of Faliszewski et al. [18] for Llull and Copeland voting by showing that even for a constant number of votes candidate control remains NP-hard. For this, we use a simple but elegant method based on the considered digraph problems.

1.5 Organization of the paper

In the following section, we give some further definitions and describe some basic observations. In Section 3, we describe our results regarding the (parameterized) complexity of the four introduced digraph problems. In Section 4, we turn our attention to voting systems. First, in Subsection 4.1, we briefly discuss some differences between candidate control in Llull and Copeland voting based on the findings of Section 3. Then, in Subsection 4.2, we investigate the complexity of candidate control in Llull and Copeland voting for a constant number of votes. Finally, in Subsection 4.3, we consider the parameterization by the "number of deleted/added candidates" for Copeland$^\alpha$ and plurality voting.

2 Preliminaries and basic observations.

Formally, an *election* (V, C) consists of a multiset V of n votes and a set C of m candidates. A *vote* is an ordered preference list, that is, a permutation of all candidates. In an election, we can either seek for a *winner*, that is, if there are several candidates who are best in the election, then all of them win, or for a *unique winner*. Note that a unique winner does not always exist. We only consider the unique-winner case, but all our results can be easily modified to work for the winner case as well. We focus on control by adding candidates (AC) or deleting candidates (DC). Then, for example, for all rational $\alpha \in [0, 1]$, we can define the decision problems of constructively controlling a Copeland$^\alpha$ election by deleting and adding candidates as follows:

CC-DC-Copeland$^\alpha$
Given: A set C of candidates, a multiset V of votes with preferences over C, a distinguished candidate $c \in C$, and an integer $k \geq 1$.
Question: Is there a subset $C' \subseteq C$ of size at most k such that c is the unique Copeland$^\alpha$ winner in the election $(V, C \backslash C')$?

CC-AC-Copeland$^\alpha$
Given: Two disjoint sets C, D of candidates, a multiset V of votes with preferences over $C \cup D$, a distinguished candidate $c \in C$, and an integer $k \geq 1$.
Question: Is there a subset $D' \subseteq D$ of size at most k such that c is the unique Copeland$^\alpha$ winner in the election $(V, C \cup D')$?

In general, the first two letters of the name of a problem stand for constructive or destructive control (CC/DC). The following two letters stand for the kind of modification (AC/DC) and are followed by the name of the considered voting system. The control problems for plurality voting and for destructive control are defined analogously (see for example [19,24]).[5] The *position* of a candidate b in a vote v is the number of candidates that are better than b in v plus one. That is, the leftmost (and best) candidate in v has position 1 and the rightmost has position m. Furthermore, within every election we fix some arbitrary order over the candidates. Specifying a subset C' of candidates in a vote means that the candidates of C' are ordered with respect to that fixed order. An occurrence of $\overleftarrow{C'}$ in a vote means that the candidates of C' are ordered in reverse order.

For an undirected graph $G = (U, E)$ and a vertex $u \in U$, the *open neighborhood* $N(u)$ of u is the set of vertices adjacent to u. Moreover, $N[u] := N(u) \cup \{u\}$ is called the *closed neighborhood* of u. For a directed graph (digraph) $D = (W, A)$ and for a vertex $w \in W$, the set of *in-neighbors* of w is defined as $N_{\text{in}}(w) := \{u \in W \mid (u, w) \in A\}$ and the set of *out-neighbors* of w is given by $N_{\text{out}}(w) := \{u \in W \mid (w, u) \in A\}$. Moreover, the *indegree* of w is defined as $d_{\text{in}}(w) := |N_{\text{in}}(w)|$ and the *outdegree* is defined as $d_{\text{out}}(w) := |N_{\text{out}}(w)|$. Furthermore, the *degree* is defined as $\deg(w) := d_{\text{in}}(w) + d_{\text{out}}(w)$. For a set of vertices $W' \subseteq W$, the *induced subgraph* $D[W']$ is the graph over the vertex set W' with arc set $\{(w, u) \in A \mid w, u \in W'\}$. In digraphs, we do not allow bidirected arcs and loops. An l-arc coloring $\mathcal{C} : A \to \{1, 2, \ldots, l\}$ is called *proper* if any two distinct arcs of the same color do not share a common vertex. A *tournament* is a digraph where, for every pair of vertices u and v, there is either (u, v) or (v, u) in the arc set.

Parameterized complexity is an (at least) two-dimensional framework for studying the computational complexity of problems [13,21,28]. One dimension is the input size n (as in classical complexity theory) and the other dimension is the *parameter* k (usually a positive integer). A problem is called *fixed-parameter tractable (FPT)* with respect to a parameter k if it can be solved in $f(k) \cdot n^{O(1)}$ time, where f is an arbitrary computable function [13,21,28]. The first two

[5] There is another version of control by adding candidates [3,17] in which one asks whether it is possible to control an election by the addition of an unlimited number of candidates. We do not consider this version here.

levels of (presumable) parameterized intractability are captured by the complexity classes W[1] and W[2]. A *parameterized reduction* reduces a problem instance (I, k) in $f(k) \cdot |I|^{O(1)}$ time to an instance (I', k') such that (I, k) is a yes-instance if and only if (I', k') is a yes-instance and k' only depends on k but not on $|I|$. If there are parameterized reductions for two problems such that each of them can be reduced to the other problem, we say that they are *FPT-equivalent*.

The following two problems are used for reductions in this work.

> HITTING SET (HS)
> *Given:* A subset family $\mathcal{F} = \{F_1, F_2, \ldots, F_m\} \subseteq 2^S$ of a base set $S = \{s_1, s_2, \ldots, s_n\}$ and an integer $k \geq 1$.
> *Question:* Is there a subset $S' \subseteq S$ of size at most k such that for every $1 \leq i \leq m$ we have $S' \cap F_i \neq \emptyset$?

The set S' is called a *hitting set*. The HITTING SET problem is known to be W[2]-complete [13]. Note that HITTING SET is NP-complete even if every subset has size two and every element occurs in exactly three subsets [22]. This restriction of HITTING SET is denoted as 3X-2-HITTING SET (3*d*-IS).

> INDEPENDENT SET (IS)
> *Given:* An undirected graph $G = (U, E)$ and an integer $k \geq 1$.
> *Question:* Is there a subset $U' \subseteq U$ of size at least k such that no two vertices in U' are adjacent?

The set U' is called an *independent set*. The INDEPENDENT SET problem is W[1]-complete on general graphs [13] and NP-complete even when restricted to graphs with maximum degree 3 (3*d*-INDEPENDENT SET) [22].

We end this section by describing the connections between the considered digraph problems and the control problems in more detail. As discussed in the introduction, there are obvious parameterized reductions from the control problems to the corresponding digraph problems with respect to the parameters number of deleted/added candidates and vertices, respectively. In the following, we show how the opposite reductions can be obtained. For this, we say that a digraph $D = (W, A)$ is *encoded* in an election (V, C) if the outcomes of the pairwise head-to-head contests reflect the arcs of the digraphs. That is, the candidate set is given by $C := \{c_i \mid w_i \in W\}$ and candidate c_i defeats candidate c_j iff $(w_i, w_j) \in D$. This can be achieved in polynomial time by a simple construction [26] as follows: For every arc $(w_i, w_j) \in D$, we add the two votes $c_i > c_j > C'$ and $\overleftarrow{C'} > c_i > c_j$ with $C' := C \setminus \{c_i, c_j\}$ to V. In these two votes, c_i beats c_j and all other pairs of candidates are tied. By this, we have a voting system with $2 \cdot |A|$ votes encoding D. The following proposition follows directly.

Proposition 1 Max-Outdegree Deletion *(*Min-Indegree Deletion*) and* CC-DC-Copeland *(*CC-DC-Llull*) are FPT-equivalent with respect to the parameters "number of deleted vertices" and "number of deleted candidates", respectively.* Max-Outdegree Addition *(*Min-Indegree Addition*) and* CC-AC-Copeland *(*CC-AC-Llull*) are FPT-equivalent with respect to the parameters "number of added vertices" and "number of added candidates", respectively.*

Finally, note that in tournaments MOD and MID coincide, since the outdegree of a vertex is exactly the number of vertices minus its indegree. Considering this from the viewpoint of the corresponding voting problems, this is fairly obvious: Copeland$^\alpha$ election differ only in the way in which ties are evaluated, and, in an election corresponding to a tournament, there is no tie between any pair of candidates.

3 Parameterized complexity of the introduced digraph problems

This section is concerned with the parameterized complexity of the four introduced digraph problems with respect to the parameterizations "number of deleted vertices" k and "maximum degree" d, for different classes of graphs. Our results for deleting vertices are summarized in Table 1 and the results for adding candidates are given in Section 3.2 (see Table 2). The next two subsection give W[2]-hardness and algorithmic results for the vertex deletion and the vertex addition problems. The W[2]-membership for all considered problems is discussed at the end of this section. Note that some of the constructions given in this section are reused in Section 4.2.

3.1 Vertex deletion

For MOD we show the following.

Theorem 1 Max-Outdegree Deletion *is W[2]-hard with respect to the parameter "number of deleted vertices" in acyclic digraphs and NP-complete in acyclic digraphs with maximum degree three.*

PROOF. Given a Hitting Set instance $H = (\mathcal{F}, S, k)$ with base set S and subset family $\mathcal{F}$, we construct the following digraph $D = (W, A)$ (see Fig. 1). The vertex set is given by $W := \{w_c\} \cup W_S \cup W_{\mathcal{F}} \cup D_w \cup \bigcup_{i=1}^{m} D_i$. Herein, w_c is the vertex we would like to become maximum degree vertex. Furthermore, we have a *subset-vertex* for every subset and an *element-vertex* for every element,

Subsets
$\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$
over elements
$S = \{s_1, s_2, \ldots, s_n\}$,
e.g., $F_1 = \{s_1, s_3\}$,
$F_2 = \{s_2, s_3\}$, ...

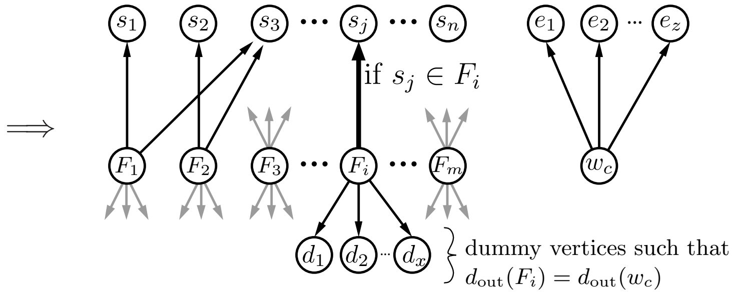

Fig. 1. Parameterized reduction from a Hitting Set-instance (left) to an MOD--instance (right).

that is, $W_{\mathcal{F}} := \{F'_i \mid F_i \in \mathcal{F}\}$ and $W_S := \{s'_i \mid s_i \in S\}$. The remaining *dummy-vertices* are specified as follows: Let z denote the maximum size over all subsets, that is, $z := max_{i \in \{1,2,\ldots,m\}} |F_i|$, then D_w consists of z vertices (needed as out-neighbors for w_c) and for every F'_i we have a (possibly empty) set D_i which contains $z - |F_i|$ further dummy-vertices. The arc set is given by $A := \{w_c\} \times D_w \cup \bigcup_{i=1}^{m} (\{F'_i\} \times D_i) \cup A_{\mathcal{F},S}$, where $A_{\mathcal{F},S}$ contains arcs from the subset-vertices to the element-vertices as follows: $A_{\mathcal{F},S} := \bigcup_{i=1}^{m} \{F'_i\} \times \{s'_j \mid s_j \in F_i\}$.

Claim: H has a hitting set of size k if and only if vertex w_c can become the only maximum outdegree vertex by deleting k vertices.

"⇒": Observe that w_c and all subset-vertices of $W_{\mathcal{F}}$ have outdegree b and all other vertices have outdegree zero. Hence, given a hitting set $S' \subseteq S$, after the deletion of the corresponding element-vertices, in the resulting graph all subset-vertices with exception of w_c have outdegree at most $b - 1$.

"⇐": Given a solution $W' \subseteq W \backslash \{w_c\}$ for the MOD-instance. If W' contains only element-vertices of W_S, then the corresponding elements constitute a hitting set: In order to make w_c the vertex with maximum outdegree we have to ensure that the outdegree of every subset-vertex of $W_{\mathcal{F}}$ is decreased by one, that is, for every subset-vertex at least one neighboring element-vertex must be included in the solution. Hence, it remains to show that we can transform any solution into a solution which consists solely of element-vertices. To this end, assume that the solution contains a dummy-vertex $d \in D_i$. Deleting d from the graph decreases only the outdegree of F'_i. Hence, we can instead delete from the graph an element-vertex s'_j with $s'_j \in F_i$, which also decreases the outdegree of vertex F'_i and has no effect on the outdegree of w_c. With a similar argument we can assume that a minimum solution does not contain any subset-vertex of $W_{\mathcal{F}}$.

The resulting digraph is acyclic (see Fig. 1), which gives the first part of the theorem. The second part follows by applying the described reduction from 3X-2-Hitting Set instead of Hitting Set. Then, since every subset contains exactly three elements we do not need any dummy-vertices and the outdegrees of the corresponding subset-vertices and the distinguished vertex are bounded by 3. Furthermore, the indegree of every element-vertex is two since every element only occurs in two subsets. Altogether, the NP-hardness for bounded degree follows. □

In contrast to the results for MOD, for MID we can state the following.

Proposition 2 Min-Indegree Deletion *can be solved in linear time in acyclic digraphs. In general digraphs, it is fixed-parameter tractable with respect to the parameter "indegree of the distinguished vertex w_c".*

PROOF. First, in a non-empty acyclic digraph there exists at least one vertex of indegree zero. Thus, the distinguished vertex w_c must have indegree zero to be the minimum indegree vertex. Hence, one can iteratively delete all other vertices with indegree zero. Using a topological ordering of an acyclic digraph, this can be done in linear time.

Second, the parameterized algorithm for MID with respect to the "indegree of the distinguished vertex" works as follows. If for an MID-instance one knows which in-neighbors of the distinguished vertex w_c are part of a minimum-cardinality solution, then the problem becomes trivial: One can delete these vertices and extend the resulting partial solution to a minimum-cardinality solution as follows. One iteratively adds all vertices of indegree smaller than the (new) indegree of w_c to the solution since all vertices of indegree smaller than the distinguished vertex must be deleted. Hence, exhaustively trying all subsets of in-neighbors of w_c yields an algorithm with running time $O(2^{d_{\text{in}}(w_c)} \cdot |W|^2)$. □

Intuitively, for the "hard" MOD problem the approach given for MID fails due to the following reason. Even in the case that we knew which neighboring vertices of the distinguished vertex w_c are part of the solution, in order to eliminate a vertex w' with higher outdegree we have to decide whether it is better to remove vertex w' or out-neighbors of it. Indeed, according to Theorem 1, MOD is NP-hard in digraphs with degree bounded by three. However, the following theorem shows that with the combined parameter maximum vertex degree d and number of deleted vertices k the problem becomes fixed-parameter tractable.

Proposition 3 Max-Outdegree Deletion *is fixed-parameter tractable with respect to the combined parameter "outdegree $d_{out}(w_c)$ of the distinguished vertex" and "number of deleted vertices k".*

PROOF. We give a simple branching strategy. If w_c is not the only vertex with maximum outdegree, then we can determine in polynomial time a vertex $u \in W \setminus \{w_c\}$ with outdegree at least $d_{\text{out}}(w_c)$. Then, to make w_c the maximum outdegree vertex, one must either delete u or an out-neighbor of u. More specifically, consider an arbitrary set $N \subseteq N_{\text{out}}(u)$ with $|N| = d_{\text{out}}(w_c)$. Clearly, if one does not delete u itself, then one has to delete at least one vertex from N. Since we do not know in advance which choice leads to a solution, we branch into all possibilities (at most $d_{\text{out}}(w_c) + 1$) to delete a vertex in $(N \cup \{u\}) \setminus \{w_c\}$. In each branch we decrease the parameter k by one (since we have deleted a vertex) and recursively solve the created subinstance. The recursion stops if w_c has become the only vertex with maximum outdegree or $k \leq 0$. For the running time note that at each level of the recursion for every subinstance we branch into at most $d_{\text{out}}(w_c) + 1$ cases and that the recursion stops at the kth level. Moreover, at every level of the recursion for every subinstance all changes are clearly doable in polynomial time. Thus, the total running time is bounded by $(d_{\text{out}}(w_c) + 1)^k \cdot |W|^{O(1)}$ □

As we will discuss in Section 4, tournaments naturally occur in the context of voting systems. Hence, in the following, we investigate the parameterized complexity of MOD restricted to tournaments. Recall that in this case MOD and MID coincide. The following theorem is based on a parameterized reduction from the W[2]-complete Dominating Set problem [13].

Dominating Set (DS)
Given: An undirected graph $G = (U, E)$ and an integer $k \geq 1$.
Question: Is there a size-k subset $S \subseteq U$ such that every vertex $u \in U \backslash S$ has a neighbor in S?

The reduction shows that MOD and MID are W[2]-hard (and NP-hard) in tournaments. Note that other prominent NP-complete problems such as Hamilton Path are polynomial-time solvable when restricted to tournaments [1].

Theorem 2 Max-Outdegree Deletion *and* Min-Indegree Deletion *are W[2]-hard with respect to the parameter "number of deleted vertices" if the input graph is a tournament.*

PROOF. We develop a parameterized reduction from Dominating Set to MOD in tournaments. The hardness for MID follows by the hardness of MOD since in tournaments both problems coincide.

The basic idea of the reduction is as follows. We construct an MOD-instance in which, aside from sets of further dummy vertices denoted by F and D, there are two copies of the vertices in the DS-instance, denoted by $\mathcal{N}$ and U', and a vertex c which we would like to become the maximum outdegree vertex. The neighborhood structure of the DS-instance is encoded in the arcs between $\mathcal{N}$ and U'. That is, we have an arc from a vertex in U' to a vertex in $\mathcal{N}$ if the respective vertices are neighbors (or the same vertex) in the DS-instance. An illustration of the resulting MOD-instance is shown in Fig. 2. Moreover, we set the arcs between all other vertices such that the following conditions are fulfilled. First, the distinguished vertex c has the same outdegree as the vertices in U'. Second, in order to decrease the outdegree of the vertices in U' below the outdegree of c by deleting k vertices, we are enforced to choose vertices from $\mathcal{N}$ such that every vertex in U' looses at least one out-neighbor. Hence, the chosen vertices correspond to a dominating set in the input instance. In the following we give the formal construction.

To simplify the proof, we assume that the graph of the DS-instance has an odd number of vertices and that $k < n$. These assumptions clearly do not limit the generality of the reduction. Given a DS-instance $(G = (U, E), k)$ where $U = \{u_1, u_2, \ldots, u_n\}$ with n odd, we construct a tournament graph $T = (W, A)$ as follows. The set of vertices is $W := \{c\} \uplus U' \uplus \mathcal{N} \uplus D \uplus F$ where c is the vertex that we would like to become maximum outdegree vertex. Furthermore,

- $U' := \{u'_i \mid i = 1, \ldots, n\}$ simulates that every vertex has to be dominated and
- $\mathcal{N} := \{n_i \mid i = 1, \ldots, n\}$ simulates that every vertex can dominate its neighborhood.
- The set $D := \{d_i \mid i = 1, \ldots, n\}$ ensures that only vertices of $\mathcal{N}$ can be deleted.
- Finally, we need a set of dummy vertices $F := \{f_i \mid i = 1, \ldots, 20n+1\}$ that are used to "set" the outdegrees of the other vertices in an appropriate way.

Next, we describe the construction of the arc set A. The goal of the construction is to ensure that c has the same outdegree as all vertices in U' and to decrease the outdegree of a vertex $u'_i \in U'$ only vertices from $\mathcal{N}$ that correspond to the closed neighborhood of u_i can be deleted. See Fig. 2 for an illustration. Within $\mathcal{N}$ we can set the arcs such that every vertex has exactly $\lfloor n/2 \rfloor$ out-neighbors inside $\mathcal{N}$ [18, Lemma 3.4]. Analogously, we set the arcs within U', F, and D. Moreover, we add the following arcs between $c, D, \mathcal{N}$, and U' to the arc set A:

- $\{c\} \times U'$ and $\{c\} \times D$ and $\mathcal{N} \times \{c\}$,
- $D \times (U' \cup \mathcal{N})$, and
- $\{u'_i\} \times \{n_j \mid u_j \in N[u_i]\}$ and $\{n_j \mid u_j \in U \setminus N[u_i]\} \times \{u'_i\}$ for $i = 1, \ldots, n$.

Finally, we describe the construction of the arcs between the dummy vertices in F and the vertices outside of F. To this end, we partition F into three sets, namely, $F_u := \{f_1, f_2, \ldots, f_{2n-1}\}$, $F_c := \{f_{2n}, f_{2n+1}, \ldots, f_{2n+\lfloor n/2 \rfloor - k}\}$, and $F_r := F \setminus (F_u \cup F_c)$. Note that $|F_u| = 2n - 1$ and $|F_c| = \lfloor n/2 \rfloor - k + 1$.

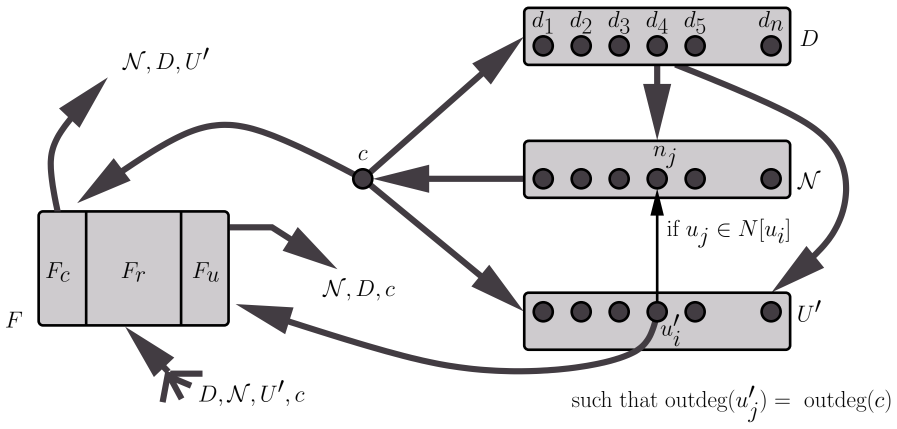

Fig. 2. Construction of the tournament in the proof of Theorem 2. The arcs between the vertices in a shaded box are allocated such that every vertex has outdegree $\lfloor n/2 \rfloor$ or $\lfloor |F|/2 \rfloor$, respectively. The bold arrows indicate bundles of arcs.

As a consequence, we have that $|F_r| > 17n$. Moreover, for $1 \leq i \leq n$ let $F_u^i := \{f_1, f_2, \ldots, f_{2n-k-|N[u_i]|+1}\}$. We add the following arcs to A.

- $\{u'_i\} \times F_u^i$ and $(F_u \setminus F_u^i) \times \{u'_i\}$ for $i = 1, \ldots, n$,
- $F_u \times (\{c\} \cup D \cup \mathcal{N})$,
- $\{c\} \times F_c$ and $F_c \times (N \cup D \cup U')$, and
- $(\{c\} \cup \mathcal{N} \cup D \cup U') \times F_r$.

This completes the construction of the tournament. By counting the out-neighbors of every vertex one can verify that the following conditions hold (herein, "$a >> b$" means that a is greater than $b + k$):

$$\begin{array}{llll} d_{\text{out}}(c) & = & 2n + \lfloor n/2 \rfloor + |F_R| - k + 1, & \\ d_{\text{out}}(u'_i) & = & 2n + \lfloor n/2 \rfloor + |F_R| - k + 1 & \text{for all } u'_i \in U', \\ d_{\text{out}}(d_i) & = & 2n + \lfloor n/2 \rfloor + |F_R| & \text{for all } d_i \in D, \\ d_{\text{out}}(n_i) & \leq & n + \lfloor n/2 \rfloor + |F_R| & \text{for all } n_i \in \mathcal{N}, \text{ and} \\ d_{\text{out}}(f) & \leq & (|F| - 1)/2 + 3n = 10n + 3n << |F_r| & \text{for all } f \in F. \end{array}$$

In particular, this means that the outdegree of c equals the outdegree of every vertex in U'. Moreover, the outdegree of a vertex in D is by $k - 1$ greater than the outdegree of c and the outdegree of every other vertex is by more than k smaller than the outdegree of c. In summary, this means that in order to make c the only vertex of maximum outdegree by the deletion of at most k vertices it remains to ensure that the outdegree of every vertex in U' and D becomes smaller than the outdegree of c. Now, we prove the correctness of the reduction.

Claim: There is a dominating set of size k iff c can become the only vertex with maximum outdegree by deleting k vertices.

"$\Rightarrow$": Let S be a dominating set of size at most k. We show that by deleting all vertices of $W' := \{n_i \in \mathcal{N} \mid u_i \in S\}$, vertex c becomes the only vertex of maximum outdegree. Clearly, the deletion of W' does not affect the outdegree of c. However, the outdegree of every $u'_i \in U'$ is decreased by at least one (by the deletion of a vertex n_j with $u_j \in N[u_i]$) and the outdegree of every $d_i \in D$ is decreased by k. Then we have $d_{\text{out}}(c) - 1 = d_{\text{out}}(d_i) \geq d_{\text{out}}(u_i) > d_{\text{out}}(n_i) > d_{\text{out}}(f_i)$ and, therefore, c is the only vertex of maximum outdegree.

"$\Leftarrow$": First, we show that every solution W' of size k for the MOD-instance contains only vertices from $\mathcal{N}$, that is, $W' \subseteq \mathcal{N}$. This relies on the fact that the difference between the outdegree of c and the outdegree of any $d_i \in D$ is exactly $k-1$. In order to make c the only vertex with maximum outdegree we have to decrease the difference for every $d_i \in D$ by every of the k deletion operations. As we cannot increase the outdegree of c, the deletion of every vertex has to decrease the outdegree of every vertex in D while it must not decrease the outdegree of c. We show that only vertices in $\mathcal{N}$ fulfill these requirements. The deletion of a vertex $x \in D \cup U' \cup F_r \cup F_c$, decreases the outdegree of c. Furthermore, the deletion of a vertex in F_u does not decrease the outdegree of a vertex in D. Hence, the only vertices whose deletion decreases the outdegree of a vertex in D and does not decrease the outdegree of c are the vertices in $\mathcal{N}$. Now, given a solution $W' \subseteq \mathcal{N}$, the deletion of W' does not affect the outdegree of c. Furthermore, for every vertex $u'_j \in U'$ at least one out-neighbor n_i must be deleted in order to ensure that the outdegree of u'_j is less than the outdegree of c. Since an out-neighbor n_i of a vertex u'_j corresponds to a vertex u_i that dominates u_j in G, the set $\{u_i \mid n_i \in W'\}$ is a dominating set in G. $\square$

3.2 Vertex addition

In the following, we describe how to obtain similar results for the digraph problems by adding vertices. Table 2 provides an overview of our results. Here, the problems seem to be even computationally harder than the deletion problems. For example, the acyclicity of the digraph does not help for solving both of the problems. Also the constructions given within the reductions are less involved (especially for the tournament case). Intuitively, this is due to the fact that one can easily "encode" much information in the subset of vertices that can be added.

Theorem 3 Min-Indegree Addition *is W[2]-complete with respect to the parameter "number of added vertices" in acyclic digraphs and NP-complete in*

Table 2

Parameterized complexity of MAX-OUTDEGREE ADDITION (MOA) and MIN-INDEGREE ADDITION (MIA). All W[2]-membership results are given in Theorem 6. The remaining results are given in [1] Theorem 3, [2] Theorem 4, [3] Proposition 4, [4] Proposition 5, [5] Theorem 5.

parameters	# deleted vertices k		maximum degree d		(k, d)	
problems	MIA	MOA	MIA	MOA	MIA	MOA
general digraphs	W[2]-c[1,5]	W[2]-c[2,5]	NP-c, $d \geq 4$[1]	FPT[3]	FPT[4]	FPT[3]
acyclic digraphs	W[2]-c[1]	W[2]-c[2]	NP-c, $d \geq 4$[1]	FPT[3]	FPT[4]	FPT[3]
tournaments	W[2]-c[5]	W[2]-c[5]	-	-	-	-

Subsets
$\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$
over elements
$S = \{s_1, s_2, \ldots, s_n\}$,
e.g., $F_1 = \{s_1, s_3\}$,
$F_2 = \{s_2, s_3\}$, ...

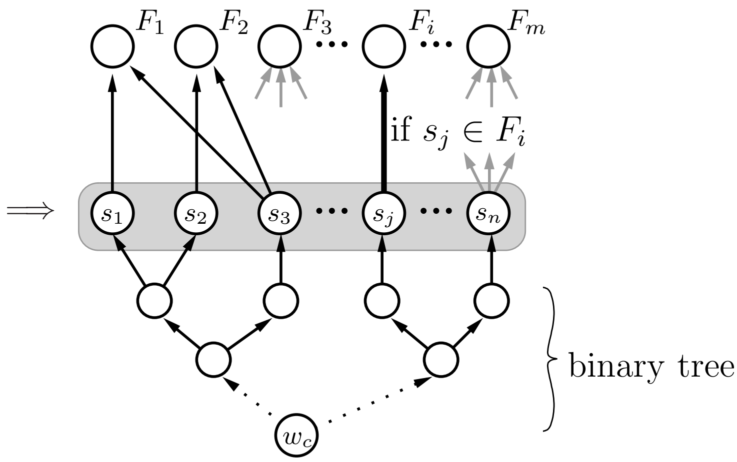

Fig. 3. Parameterized reduction from a HITTING SET-instance (left) to an MIN-INDEGREE ADDITION-instance (right).

acyclic digraphs with maximum degree four.

PROOF. The theorem is base on a reduction from HITTING SET. The construction is illustrated in Fig. 3. Herein, the vertices that can be added are marked by a shaded box. These vertices correspond to the elements in S. The distinguished vertex w_c and all the "subset-vertices" F_i have indegree zero before the addition of any "element-vertex." All other vertices have indegree at least one. Especially the binary tree consists of dummy vertices that ensure that each s_i has indegree at least 1. As discussed below, the binary tree structure is useful for the bounded-degree case. Adding an element-vertex s_j in the digraph has the effect that for all subset-vertices corresponding to the subsets containing s_j the indegree is increased above the indegree of the distinguished vertex w_c, that is, the corresponding subsets are "hit" in the HITTING SET-instance. Hence, a hitting set one-to-one corresponds to a solution of the constructed MIN-INDEGREE ADDITION-instance. Clearly, the constructed graph is acyclic and the first part of the theorem follows. To see the NP-hardness

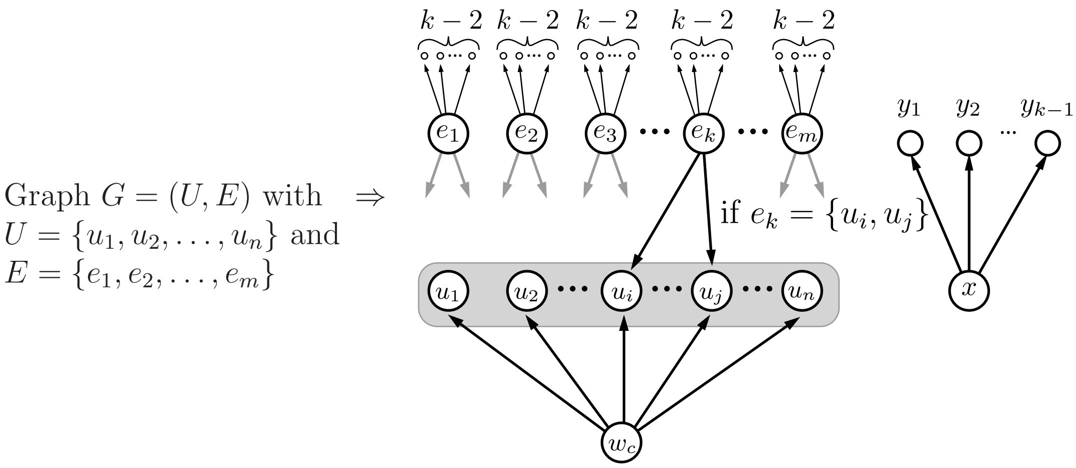

Fig. 4. Parameterized reduction from Independent Set (left) to Max-Outdegree Addition (right).

for $d \geq 4$, we reduce from 3X-2-Hitting Set. Then, the constructed graph has maximum degree four. More precisely, every element-vertex has at most three out-going and one in-going arcs, that is, degree four, and all other vertices have degree at most two. This settles the second statement of the theorem. □

Theorem 4 Max-Outdegree Addition *is W[2]-hard with respect to the parameter "number of added vertices" in acyclic digraphs.*

PROOF. This theorem follows by a reduction from Independent Set. The construction is given in Fig. 4. Herein, the vertices that can be added are marked by a shaded box. These vertices correspond to the vertices of the graph of the IS-instance. Each such vertex u_i has in-going arcs from the distinguished vertex w_c and from an "edge-vertex" e_k if u_i is incident to e_k. Moreover, every edge-vertex has $k-2$ further out-neighbors and there exists a vertex x with outdegree $k-1$. Note that before the addition of any vertex the outdegree of the distinguished vertex w_c is zero. Hence, in order to increase the outdegree of w_c above the outdegree of x one has to add at least k vertices. However, for every edge-vertex (that has degree $k-2$ before the addition of any vertex) one is allowed to add at most one of its two out-neighbors to ensure that its outdegree does not exceed $k-1$. Hence, an independent set of size k one-to-one corresponds to a set of vertices whose addition makes w_c the only vertex of maximum outdegree.

In the given reduction the distinguished vertex has unbounded outdegree. Parameterized by the outdegree of the distinguished vertex MOA becomes fixed-parameter tractable.

Proposition 4 Max-Outdegree Addition *is fixed-parameter tractable with respect to the parameter "outdegree of the distinguished vertex* w_c*."*

PROOF. We show that a minimum-size solution W' contains only out-neighbors of the distinguished vertex w_c. Assume that W' contains a vertex x that is not an out-neighbor of w_c. Then, deleting x from W' does not decrease the outdegree of w_c and, obviously, does not increase the outdegree of any other vertex. That is, w_c remains the vertex with maximum outdegree and W' cannot have minimum size. Hence, one can enumerate all possible subsets of $N_{\text{out}}(w_c)$ checking whether the current set forms a valid solution. Since the number of subsets of $N_{\text{out}}(w_c)$ is $2^{|N_{\text{out}}(w_c)|}$, fixed-parameter tractability follows directly. □

Concerning MIA, which is NP-hard on graphs with degree at least four, we show fixed-parameter tractability for the combined parameter "maximum indegree" and "number of added candidates."

Proposition 5 Min-Indegree Addition *is fixed-parameter tractable with respect to the combined parameter "maximum indegree" and "number of added candidates".*

PROOF. Consider an MIA-instance. If there exists a vertex v with indegree smaller than the indegree of the distinguished vertex, one has to add at least one in-neighbor of v. Since the number of in-neighbors is bounded we can branch into all possible choices of adding an in-neighbor. In each case we can decrease parameter k by one. With the same argument as in the proof of Proposition 3 this leads to the running time $d_{\text{in}}^k \cdot |W|^{O(1)}$, where d_{in} denotes the maximum indegree. □

The W[2]-hardness proof for MOA/MIA restricted to tournaments is less involved than the proof for MOD/MID. It can be achieved by using the basic idea of Theorem 3 combined with a similar but less complicated construction of dummy candidates as the one that is used in the reduction for Theorem 2.

Theorem 5 Max-Outdegree Addition *and* Min-Indegree Addition *are W[2]-hard with respect to the parameter "number of deleted vertices" even in the case that the input graph is a tournament.*

We conclude the considerations on the four digraph problems by showing their containment in W[2]. We use an alternative characterization of W[2], called W*[2], introduced by Downey and Fellows [12] who showed that W*[2]=W[2]. To explain this concept, we need some definitions in the context of Boolean circuits. We distinguish two types of gates: *Large gates* are $\vee$ gates and $\wedge$ gates with unrestricted fan-in. *Small gates* are $\neg$ gates, $\vee$ gates, and $\wedge$ gates with bounded fan-in. In the "traditional" characterization of W[2] the fan-in of a small gate has to be bounded by a constant, whereas in the characterization used here it is sufficient if the fan-in of a small gate is bounded by a function of the considered parameter. The *depth* of a circuit C is the maximum number of gates on an input-output path in C. The *weft* of a circuit C is the maximum number of large gates on an input-output path in C. The k-Weighted Circuit Satisfiability (k-WCS) problem has as input a circuit C and a positive integer k, and asks whether C has a weight-k satisfying assignment (an assignment setting the values of exactly k input gates to 1). In the proof of the following theorem, we use that a parameterized problem is in W[2] if it is reducible to k-WCS for a family of circuits $\mathcal{C}$ satisfying the following two conditions [12]:

(1) The weft of any circuit $C \in \mathcal{C}$ is at most two where any gate with fan-in bounded by an arbitrary function of k is considered small.
(2) The depth of any circuit is at most $h(k)$ for an arbitrary function h.

With this machinery, we can show the following theorem.

Theorem 6 Max-Outdegree Deletion*,* Max-Outdegree Addition*,* Min-Indegree Deletion*, and* Min-Indegree Addition *are in W[2].*

PROOF. First, we show the W[2]-membership for MOD by reducing it to the special case of k-WCS fulfilling Conditions (1) and (2). Let $(D = (W, A), w_c, k)$ denote an MOD-instance. If there is a vertex $w \in W$ with $d_{\text{out}}(w) \geq d_{\text{out}}(w_c)+k$, the only possibility to solve MOD is to delete w. Thus, we can assume that no such vertex exists.

The basic idea of the reduction is analogous to the proof of [12, Theorem 1]: The input variables correspond to k copies of the vertex set $W\backslash\{w_c\}$, more specifically, there are k variables for every vertex of $W\backslash\{w_c\}$. Furthermore, the construction ensures that exactly one variable of every copy of the vertex set must be set to true, that is, one "chooses" exactly k vertices (one of each copy) for the solution. Roughly speaking, this construction is useful since it enables us to "select" a subset of the chosen vertices by selecting a subset of the copies of the vertex set which are 2^k possibilities (a function only depending on k)

instead of selecting a subset of at most k vertices out of W (whose size may not be bounded by a function of k).

Formally, the set of variables is $X := \{c[i,w] \mid 1 \leq i \leq k, w \in W \backslash \{w_c\}\}$. Herein, $c[i,w] = 1$ means that w is the selected vertex of the ith copy of the vertex set. Furthermore, for an integer $r \leq k$, let $S(k,r)$ denote the set of all r-element subsets of $\{1,\ldots,k\}$.

The deletion of vertices from D can affect the outdegree of w_c. In the following formulation, we try all possible amounts by which the outdegree of w_c can be decreased. Let the number by which the outdegree of w_c is decreased be s. Then, for every other vertex $w \in W\backslash\{w_c\}$ one has that the outdegree of w must be decreased at least by $u_{w,s} = d_{\text{out}}(w) - d_{\text{out}}(w_c) - s + 1$. Note that since we assume $d_{\text{out}}(w) < d_{\text{out}}(w_c) + k$ it holds that $u_{w,s}$ is at most k. With these definitions, we can formulate a family of circuits as follows:

$$\bigvee_{s\in\{0,\ldots,k\}} ((\bigvee_{J\in S(k,k-s)} \bigwedge_{j\in J} \bigvee_{w\notin N_{\text{out}}(w_c)} c[j,w])\wedge \quad (1a)$$

$$(\bigwedge_{w\in W\backslash\{w_c\}}((\bigvee_{j\in\{1,\ldots,k\}} c[j,w])\vee \quad (1b)$$

$$(\bigvee_{J\in S[k,u_{w,s}]} \bigwedge_{j\in J} \bigvee_{w'\in N_{\text{out}}(w)} c[j,w'])))) \quad (1c)$$

$$\wedge \quad (\bigwedge_{i\in\{1,\ldots,k\}} \bigwedge_{w\neq w'}(\neg c[i,w] \vee \neg c[i,w'])) \quad (2)$$

$$\wedge \quad (\bigwedge_{w\in W\backslash\{w_c\}} \bigwedge_{i\neq j}(\neg c[i,w] \vee \neg c[j,w])) \quad (3)$$

First, we argue that the circuits work correctly. The gates of (2) ensure that at most one vertex of every copy of the vertex set is selected and the gates of (3) ensure that every vertex is selected in at most one copy of the vertex set. The first part of the gates checks whether there is a solution for any possible outdegree which w_c can have after deleting the vertices. For this, recall that s is the amount by which the outdegree of w_c is decreased: In (1a) "the expression" becomes true if there is a size-$(k-s)$-subset of indices such that all vertices that are selected for the corresponding indices are not in $N_{\text{out}}(w_s)$. Hence, $k-s$ of the deleted vertices are not out-neighbors of w_c and, thus, the outdegree of w_c after deleting the k vertices (for which $c[i,v]$ is true) is at least $d_{\text{out}}(w_c) - s$. The gates of (1b) and (1c) ensure that for every vertex $w \in W\backslash\{w_c\}$ either w is deleted or its outdegree in the resulting instance is smaller than the final outdegree of w. More precisely, there must be either an index j for which $c[j,w]$ is true (1b) or there must be a subset of indices of size $u_{w,s}$ such that the corresponding deleted vertices are out-neighbors of w (1c). Hence, if there is a weight-k satisfying assignment, then after deleting the set of vertices $\{w \in W \mid \exists j \in \{1,\ldots,k\}$ with $c[j,w] = 1\}$ vertex w_c has the maximum outdegree in D.

Second, we consider the size of the circuits. Regarding the weft, the only gates with unbounded fan-in are the $\bigwedge$-gates over all vertices $w \in W\backslash\{w_c\}$ and the $\bigvee$-gates over the out-neighbors of a vertex. It is easy to check that there are at most two gates of this type at one input-output path. The depth of the circuit is obviously bounded by a constant. Thus, MOD is contained in W[2].

For the other three problems one can use the same methods to show the membership in W[2]. For the vertex addition problems the variable set contains only copies of all vertices that can be added. Then, for MOA, MID, and MIA one can adapt the first part of the constraints in a straightforward manner. □

4 Parameterized complexity of candidate control

In this section, we come back to voting systems. The digraph problems considered in the previous section turn out to be very useful to determine the parameterized complexity of candidate control for different voting systems. In Subsection 4.1, we briefly discuss some consequences of the results obtained for the digraph problems for control in Llull and Copeland voting. In Subsection 4.2, we show NP-hardness for candidate control in Llull and Copeland voting for a constant number of votes. Finally, in Subsection 4.3, we provide parameterized intractability results with respect to the "number of deleted/added candidates" for plurality and Copeland$^\alpha$ voting.

4.1 Llull and Copeland voting

The only difference between Llull and Copland voting is the way in which ties are evaluated. If two candidates are tied in their head-to-head contest, both of them get zero points in a Copeland election and one point in a Llull election. As stated by Faliszewski et al. [17], the different evaluation of ties can make the dynamics of Llull's system quite different from those of Copeland's system. For example, they observed that the proof techniques used to show NP-hardness are quite different for different ways of evaluating ties. However, for the problems considered in this work, until now, there was no measurable difference in the computational complexity of candidate control between Llull and Copeland voting. Using a two-dimensional view on the problems, we identify cases in which their complexities differ. As CC-DC-Llull/CC-AC-Llull and CC-DC-Copeland/CC-AC-Copeland are FPT-equivalent to MID/MIA and MOD/MOA, respectively, all results of Tables 1 and 2 also hold for them. In particular, the bounded-degree scenario for MID (Proposition 2) seems to be fairly realistic: To control an election is particularly attractive if the distinguished candidate is already "close" to be a winner. A natural indicator for "closeness" is the number of candidates that beat the distinguished candidate. That is, the corresponding distinguished vertex has bounded indegree. In this case, in contrast to Copeland elections, Llull elections are "easy" to control. Thus, as a direct consequence of Theorem 1 and Proposition 2, we state the following.

Corollary 1 *CC-DC-Llull is fixed-parameter tractable with respect to the parameter "number of candidates defeating the distinguished candidate". CC-DC-Copeland is NP-hard to control even if for every candidate the number of candidates that are not tied with it is at most three.*

4.2 Number of votes as parameter

In many election scenarios there is only a small number of votes. For example, consider a human resources department where few people are deciding which job applicant gets the employment. Another prominent example is rank aggregation. An open question of Faliszewski et al. [18] regards the parameterized complexity of candidate control in Copeland$^{\alpha}$ voting with respect to the parameter "number of votes". We answer this question for the two important special cases Llull and Copeland by making use of the corresponding digraph problems. More precisely, we devise four reductions showing that the problems of controlling Llull and Copeland voting by deleting or adding candidates are NP-complete even in the case of a constant number of votes. Each reduction is from a special case of the corresponding digraph problem. For all but one reduction the NP-hardness of the considered special case follows from reductions given in Section 3. The new part of the reductions is to show how a given instance of the digraph problem can be encoded into an election using a constant number of votes. Recall that, as discussed in Section 2, we say that a digraph encodes an election if the outcomes of the pairwise head-to-head contests reflect the arcs in the digraphs. That is, if there is an arc from vertex v to vertex w, then the corresponding candidate v must be better than w in more than half of the votes. Here, the encoding of all digraphs into a constant number of votes is based on the idea to partition the set of arcs into a constant number of subsets in a way such that each subset can be encoded independently of the others by each time two votes. [6] A useful tool to obtain such partitionings are arc colorings for digraphs.

Lemma 1 *If there is a proper ℓ-arc coloring for a digraph D, then D can be encoded into 2ℓ votes.*

PROOF. Given a digraph $D = (V, A)$ and a proper ℓ-arc coloring $\mathcal{C} : A \to \{\mathcal{R}_1, \mathcal{R}_1, \ldots, \mathcal{R}_\ell\}$ for D. In the underlying undirected graph of D the edges of

[6] In contrast, in previous works, as for example [17], only one arc was encoded into two votes.

the same color class form a matching, that is, two arcs of the same color do not share a common vertex. Hence, the coloring $\mathcal{C}$ partitions the arc set into ℓ classes of independent arcs. We next describe how the arcs of graph D can be encoded in an election with 2ℓ votes. Let $A_{\mathcal{R}_1} = \{(r_1, r_{1'}), \ldots, (r_q, r_{q'})\}$ denote the arcs colored by $\mathcal{R}_1$. Furthermore, $W_{\overline{\mathcal{R}_1}}$ denotes the set of vertices that are not incident to any arc of $A_{\mathcal{R}_1}$. To encode $A_{\mathcal{R}_1}$, we add the two votes

$$v_{\mathcal{R}_1,1} : r_1 > r_1' > r_2 > r_2' > \cdots > r_q > r_q' > W_{\overline{\mathcal{R}_1}}$$

$$v_{\mathcal{R}_1,2} : \overleftarrow{W_{\overline{\mathcal{R}_1}}} > r_q > r_q' > \cdots > r_2 > r_2' > r_1 > r_1'$$

to the election. In the same way, for each $1 < i \leq \ell$ we add two votes $v_{\mathcal{R}_i,1}$ and $v_{R_i,2}$ for the arcs colored by $\mathcal{R}_i$. The correctness of the construction follows from two observations. First, since the arcs of the same color do not share common endpoints, in every vote all vertices occur exactly once and we have a valid election. Second, consider an arc $(w', w'') \in A$ with $\mathcal{C}((w', w'')) = R_i$ for some $1 \leq i \leq \ell$. Then, w' defeats w'' in the votes $v_{\mathcal{R}_i,1}$ and $v_{\mathcal{R}_i,2}$ and ties with w'' in the remaining votes. Moreover, since every arc occurs in exactly one color class, all arcs are encoded, and, since all other candidates are tied in all pairs of the votes, we have ties between all other pairs of candidates. □

Every undirected graph admits a proper arc/edge-coloring using $\Delta + 1$ colors, where Δ denotes the maximum degree. Moreover, Δ is a lower bound on the number of colors that are necessary for any proper arc/edge coloring. For arbitrary graphs, it is NP-complete to decide whether the graph has an proper Δ-arc/edge coloring. In contrast, by König's Theorem, for all bipartite graphs one can find a proper Δ-arc/edge coloring in polynomial time [25].

Lemma 2 *(König [1916]) A bipartite graph is Δ-edge-colorable, where Δ denotes the maximum degree of the graph. A corresponding proper Δ-edge coloring can be computed in polynomial time.*

These two lemmas are used to show the following.

Theorem 7 *Controlling Llull and Copeland by deleting/adding candidates is NP-complete for a constant number of votes. More precisely,* CC-DC-COPELAND *is NP-complete for six votes,* CC-AC-COPELAND *is NP-complete for eight votes,* CC-DC-LLULL *is NP-complete for ten votes, and* CC-AC-LLULL *is NP-complete for eight votes.*

PROOF. For all problems NP-membership is obvious. We start with the NP-hardness proof for CC-DC-COPELAND to demonstrate the basic idea. Consider the reduction from the NP-complete 3X-2-HITTING SET to MOD as depicted in Fig. 1. The digraph D of a resulting MOD-instance (D, w_c, k) has maximum degree three and the underlying undirected graph of D is bipartite. More precisely, one partition consists of the subset-vertices and w_c, and the other partition consists of the element-vertices and the neighbors of w_c. Note that as we reduce from 3X-2-HITTING SET, we do not have any further dummy vertices. It follows directly from Lemma 2 that D has a proper 3-arc coloring. Thus, by Lemma 1, D can be encoded into an election of six votes resulting in an equivalent instance of CC-DC-COPELAND.

Next, we argue that CC-AC-Llull is NP-complete for eight votes. According to Theorem 3 MIA is NP-complete even when restricted to graphs with maximum degree four. Moreover, observe that the underlying undirected graph of the digraph constructed in the respective reduction from 3X-2-Hitting Set (see Fig. 3) is bipartite. Hence, for CC-AC-Llull the NP-hardness follows in complete analogy to CC-DC-Copeland by using Lemmas 1 and 2.

For CC-AC-Copeland, we show how to encode an NP-hard MOA-instance that results from the reduction of $3d-$INDEPENDENT SET (Fig. 4) into an election of eight votes. Note that since MOA is fixed-parameter tractable with respect to the maximum degree, it is polynomial-time solvable for an instance with constant degree. Hence, we can not assume that the maximum degree in the constructed MOA-instance is constant. However, by using the reduction from $3d$-INDEPENDENT SET ($3d$-IS) we can still make use of a degree restriction of the subgraph induced by $\{e_i \mid i = 1, \ldots, m\} \cup \{u_j \mid j = 1, \ldots, n\}$. Since the degree within this subgraph is at most three and its underlying undirected graph is bipartite, due to Lemmas 1 and 2 it can be encoded into six votes. The remaining arcs can be encoded into two further votes as follows. Let $S(e_i)$ denote the $k-2$ dummy out-neighbors of e_i, then we can add the following two votes
$e_1 > S(e_1) > \cdots > e_m > S(e_m) > w_c > u_1 > \cdots > u_n > x > y_1 > \cdots > y_{k-1}$
$x > y_1 > \cdots > y_{k-1} > w_c > u_n > \cdots > u_1 > e_m > \overleftarrow{S(e_m)} > \cdots > e_1 > \overleftarrow{S(e_1)}$.
This completes the proof for CC-AC-Copeland.

Finally, we show that CC-DC-Llull is NP-hard for ten votes. We present a reduction from $3d$-IS to MID and show that the resulting MID-instance can be encoded into an election with ten votes. Note that since CC-DC-Llull is solvable in polynomial time on acyclic digraphs and FPT with respect to the degree, in contrast to the other problems, there is no previous reduction we can reuse.

Given a $3d$-IS-instance consisting of an undirected graph $G = (U, E)$ with $n := |U|$ and a non-negative integer k, we construct an MID-instance consisting of a graph $D = (W, A)$, a distinguished vertex w_c, and a non-negative integer k. See Fig. 5 for an illustration. The vertex set W is the disjoint union of the following sets:

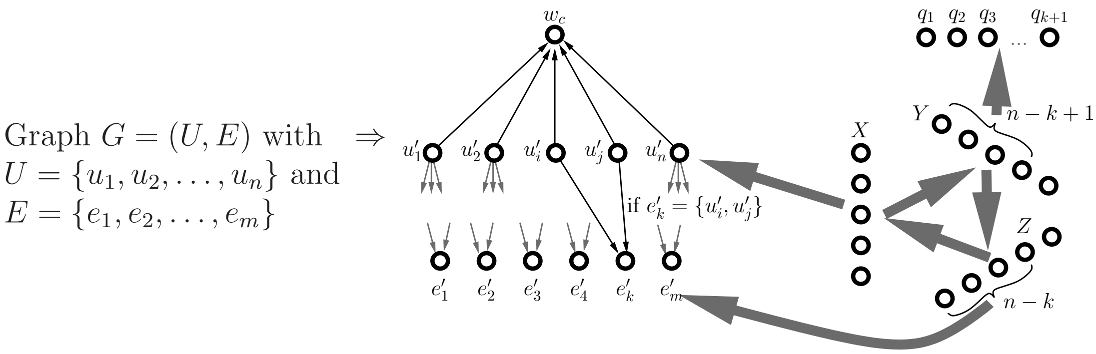

Fig. 5. Reduction from an 3d-Independent Set-instance (left) to an Min-Indegree Deletion-instance (right).

- $\{w_c\}$, the distinguished vertex,
- $U' := \{u'_i \mid u_i \in U\}$, one new vertex for every IS-vertex,
- $E' := \{e'_i \mid e_i \in E\}$, one new vertex for every IS-edge,
- three sets of dummy vertices $X := \{x_1, x_2, \ldots, x_n\}$, $Y := \{y_1, y_2, \ldots, y_n\}$, $Z := \{z_1, z_2, \ldots, z_n\}$ that are needed to "set" the indegrees of the other vertices in an appropriate way, and
- $Q := \{q_1, q_2, \ldots, q_{k+1}\}$, vertices that enforce that the indegree of the distinguished vertex must decrease by k.

The basic idea is to set the arcs such that the indegree of the distinguished vertex w_c has to be decreased by k. Furthermore, to decrease the indegree of w_c one can only delete vertices that correspond to vertices of the 3d-IS-instance, that is, vertices of U'. The deletion of such a vertex does not only decrease the indegree of w_c but also the indegree of the (at most three) vertices that correspond to its incident edges. By using the dummy vertices one can set the indegrees of the edge-vertices such that one can delete at most one neighbor of any edge-vertex. Then, to make w_c a winner one has to delete k vertices that correspond to vertices of the 3d-IS-instance such that for every edge at most one of its incident vertices is deleted. Thus, the deleted vertices must correspond to an independent set. In the following, we describe the arc set A that is given by the union of the following disjoint arc sets.

- $A_{U',w_c} := U' \times \{w_c\}$,
- $A_{U',E'} := \bigcup_{u_j \in U}(\{u'_j\} \times \{e'_i \mid e_i \in E \wedge u_j \cap e_i \neq \emptyset\})$,
- $A_{X,Y,Z} := (X \times Y) \cup (Y \times Z) \cup (Z \times X)$,
- $A_{X,U'} := X \times U'$,
- $A_{Y,Q} := \{y_1, y_2, \ldots, y_{n-k+1}\} \times Q$, and
- $A_{Z,E'} := \{z_1, z_2, \ldots, z_{n-k}\} \times E'$.

For an illustration of the construction see Fig. 5. Note that every vertex e'_i has exactly two in-going arcs from U' and $n - k$ from Z. Hence, it can easily be verified that in D the indegree for all $e'_i \in E'$ is $n - k + 2$, the indegree of q_i is $n - k + 1$ for $i = 1, \ldots, k$, and the indegree of all remaining vertices is n.

We next prove the correctness of the reduction. Let $I \subseteq U$ be an independent set of G. After deleting $S := \{u'_i \mid u_i \in I\}$ from D we have $d_{\text{in}}(w_c) = n - k$ and since I is an independent set the indegree of every vertex $e'_i \in E'$ is decreased by at most one, that is, $d_{\text{in}}(e'_i) \geq n - k + 1$. The indegree of all other vertices is not affected. Therefore, w_c is the vertex with minimum indegree.

Let $S \subseteq W$ be an optimal solution for MID. We can assume that $S \subseteq U'$, since in order to improve w_c against the vertices from Q we must delete k vertices of $N_{\text{in}}(w_c)$. This is due to the fact that we cannot delete all $k+1$ vertices from Q. Since S contains only vertices from U', the indegree of w_c is exactly $n - k$. Moreover, for every $e_i = \{u_j, u_k\} \in E$ in order to ensure that $d_{\text{in}}(e'_i) > n - k$ we can have either u'_j or u'_k in the solution. Hence, $\{u_i \in U \mid u'_i \in S\}$ is an independent set.

In the remainder of this proof we show that the graph D can be encoded into an election using ten votes in total. Because G has maximum degree 3, it is easy to see that the underlying undirected graph of $D[U' \cup E']$ is bipartite and has maximum degree three. Consequently, following Lemma 2, there exists a proper 3-arc coloring for $D[U' \cup E']$ and the information for this subgraph can be encoded into six votes (Lemma 1). Let $R := A \backslash \{X, Y, Z\}$. The arcs between X, Y, and Z can be encoded into the following three pairs of votes.

(1) $X > Y > Z > R$ and $\overleftarrow{R} > \overleftarrow{X} > \overleftarrow{Y} > \overleftarrow{Z}$.
(2) $Y > Z > X > R$ and $\overleftarrow{R} > \overleftarrow{Y} > \overleftarrow{Z} > \overleftarrow{X}$.
(3) $Z > X > Y > R$ and $\overleftarrow{R} > \overleftarrow{Z} > \overleftarrow{X} > \overleftarrow{Y}$.

Since the arcs between X, Y, and Z are independent from the arcs in $D[U' \cup E']$, the encoding of both sets of arcs can be done within the same three pairs of votes. It remains to encode the arcs from Y to Q, the arcs from X to U', the arcs from Z to E', and the arcs from U' to w_c. It is not hard to see that this can be done by using two further pairs of votes. □

4.3 Number of deleted/added candidates as parameter

To control an election without raising suspicion one may add or delete only a limited number of candidates. Here, we investigate whether it is possible to obtain efficient algorithms under this assumption. More specifically, we consider the parameterized complexity of destructive and constructive control by adding or deleting a fixed number of candidates. Our results are summarized in Table 3. It turns out that all NP-complete problems are intractable from this parameterized point of view as well. This even holds true for plurality voting, which can be considered as the "easiest" voting system in terms of winner evaluation and for which the Manipulation problem can be solved optimally

Table 3
Results in boldface are new. The results for Copeland$^\alpha$ hold for all $0 \leq \alpha \leq 1$. The W[2]-hardness results for CC-AC-Plurality and DC-AC-Plurality follow from the NP-completeness proofs [3,24]. The polynomial-time (P) results are from [17,18].

	Copeland$^\alpha$		plurality	
	CC	DC	CC	DC
Adding Candidates (AC)	**W[2]-c**	P	W[2]-h	W[2]-h
Deleting Candidates (DC)	**W[2]-c**	P	**W[2]-h**	**W[1]-h**

by a simple greedy strategy [11]. Whereas the results for Copeland$^\alpha$ voting can be obtained easily from the results of the corresponding digraph problems, we give two reductions with new ideas for constructive and destructive control in plurality voting.

4.3.1 Copeland$^\alpha$

Having no ties in the pairwise head-to-head contests between all pairs of candidates is a realistic scenario. It is always the case for an odd number of votes and likely for a large number of votes. Thus, it is interesting to investigate this setting. Note that the NP-hardness proofs of candidate control in Copeland$^\alpha$ voting rely on ties [17,18]. For elections without ties in all pairwise head-to-head contests, CC-DC-Copeland$^\alpha$, as well as CC-AC-Copeland$^\alpha$, coincide for all $0 \leq \alpha \leq 1$, since these problems only differ in the way ties are evaluated.

As discussed in the introduction, MOD/MOA and CC-DC-Copeland/CC-AC-Copeland are FPT-equivalent. Using the same reductions one can show that MOD/MOA in tournaments are FPT-equivalent to CC-DC-Copeland$^\alpha$/CC-AC-Copeland$^\alpha$ without ties. Thus, we obtain the following corollary from Theorem 2 and Theorem 5.

Corollary 2 *For a tie-free voting and $0 \leq \alpha \leq 1$, CC-DC-Copeland$^\alpha$ is W[2]-complete with respect to the "number of deleted candidates" and CC-AC-Copeland$^\alpha$ is W[2]-complete with respect to the "number of added candidates".*

4.3.2 Plurality

In this section, we consider plurality voting and show that candidate control is not only NP-hard but also intractable from parameterized point of view. Note that the class containment in W[1] or W[2] for all kinds of candidate control in plurality voting is open.

For plurality voting, the W[2]-hardness results for control by adding candidates follow from existing NP-hardness proofs [3,24]. Hence, we can state the following theorem.

Theorem 8 *Destructive and constructive control of plurality voting by adding candidates are W[2]-hard with respect to the "number of added candidates".*

In contrast, the reductions used to show NP-hardness for destructive and constructive control by deleting candidates [3,24] do not imply their W[1]-hardness. Thus, we develop new parameterized reductions. For the constructive case we can show W[2]-hardness by a reduction from MOD. Note that the encoding of an MOD instance into a plurality election is more demanding than for Copeland voting and the other direction (encoding a plurality election into an MOD instance) is not obvious. Therefore, in contrast to the considerations for Copeland$^\alpha$ elections, where the main focus was on showing the W[2]-hardness of MOD on tournaments, here the technical part is the reduction from MOD to CC-DC-Plurality itself. Recall that for control in plurality our input consists of preference lists and the score of a candidate is the number of its first positions.

Theorem 9 *Constructive control of plurality voting by deleting candidates is W[2]-hard with respect to the parameter "number of deleted candidates".*

PROOF. We present a parameterized reduction from MOD. The basic idea is to construct a plurality election such that, for every vertex w of the MOD-instance with higher outdegree than the distinguished vertex w_c, the corresponding candidate w' has a higher plurality score than the distinguished candidate c. More precisely, the difference between the score of w' and the score of c equals the difference of their outdegrees, that is, $\text{score}(w') - score(c) = d_{\text{out}}(w') - d_{\text{out}}(c)$. Furthermore, due to our construction there are only two possibilities to make c to beat w' in the plurality election. First, one can delete w' itself. Second, the deletion of a candidate corresponding to an out-neighbor of w decreases the score of w' by one point but the score of c remains unchanged. Thus, in this case, one has to delete at least $d_{\text{out}}(w') - d_{\text{out}}(c) + 1$ candidates that correspond to out-neighbors of w'. In both cases the deletion of the corresponding vertices in the MOD-instance has the effect that the distinguished vertex has higher outdegree than w. In the following, we describe the formal construction.

Given an MOD instance $(D = (W, A), w_c, k)$ with $W = \{w_1, w_2, \dots, w_n\}$ and $w_c = w_1$, we construct an election (V, C) as follows: We have one candidate corresponding to every vertex, that is, $C' := \{c_i \mid w_i \in W\}$. The set of candidates C then consists of C' and an additional set F of "dummy" candidates (only used to "fill" positions that cannot be taken by other candidates in our construction). The multiset of votes V consists of two subsets V_1 and V_2.

In V_1, for every $c_i \in C'$ we have $d_{\text{out}}(w_i)$ votes in which c_i is at the first position and with dummy candidates in the positions from 2 to $k+1$. Then, for every such vote, the remaining candidates follow in arbitrary order. In V_2, for every $c_i \in C'$ we have $|W|$ votes in which c_i is at the first position. For all candidates $c_j \neq c_i$ with $w_j \notin N_{\text{in}}(w_i)$, we ensure that in exactly one of these $|W|$ votes c_j is at the second position. In all other of these votes, the second position is filled with a dummy candidate. Moreover, we add dummies to all positions from 3 to $k+1$. Concerning the dummies, in V_1 and V_2 we ensure that every dummy candidate $f \in F$ has a position better than $k+2$ only in one of the votes. This can be done by using a different dummy candidate for every position. Obviously the size of F is less than $(k+1) \cdot |V|$. The dummies exclude the possibility of "accidently" getting candidates in the first position. Note that by deleting k candidates only a candidate that is at one of the first $k+1$ positions in a vote has the possibility to increase its plurality score. Furthermore, by construction, the dummy candidates fulfill the following two conditions. First, the score of a dummy candidate can become at most one. Second, it does never make sense to delete a dummy as by this only other dummies can get into the first position of a vote. Next, we prove the correctness of the reduction.

Claim 1 *Candidate c_1 can become the plurality winner of (V, C) by deleting k candidates iff w_1 can become the only maximum-degree vertex in D by deleting k vertices.*

"$\Rightarrow$": Denote the set of deleted candidates by R. We show that after deleting the set of vertices $W_R := \{w_i \mid c_i \in R\}$ the vertex w_1 is the only vertex with maximum degree. Before deleting any candidates, for every candidate c_i we have $\text{score}(c_i) = \text{score}(c_1) + s_i$ with $s_i := d_{\text{out}}(w_i) - d_{\text{out}}(w_1)$. After deleting the candidates in R, candidate c_1 is the winner. Hence, for $i = 2, \ldots, |W|$ we must have either that $\text{score}(c_i) < \text{score}(c_1)$ or that c_i is deleted. For a non-deleted candidate c_i with $i > 1$ the difference between $\text{score}(c_i)$ and $\text{score}(c_1)$ must be decreased by at least $s_i + 1$. By construction, the only way to decrease the difference by one is to delete a candidate such that c_1 becomes first in one more vote and c_i does not increase the number of its first positions. All candidates that can be deleted to achieve this correspond to vertices in $N_{\text{in}}(w_i) \setminus N_{\text{in}}(w_1)$. To improve upon c_i we must delete at least $s_i + 1$ candidates that fulfill this requirement. Hence, in D the outdegree of w_i is reduced to be less than the outdegree of w_1.

"$\Leftarrow$": Let $T \subseteq D$ denote the solution for MOD. We can show in a straightforward way ("reverse" to the other direction) that by deleting the set of candidates $C_T := \{c_i \mid w_i \in T\}$ candidate c_1 becomes a plurality winner. □

In contrast to Copeland$^\alpha$ voting, for plurality voting destructive control by deleting candidates is NP-hard [24]. We show that it is even W[1]-hard by presenting a parameterized reduction from the W[1]-complete CLIQUE problem [13]. Given an undirected graph $G = (W, E)$ and a positive integer k, the CLIQUE problem asks to decide whether G contains a complete subgraph of size at least k.

Theorem 10 *Destructive control of plurality voting by deleting candidates is W[1]-hard with respect to the parameter "number of deleted candidates".*

PROOF. Given a CLIQUE instance $(G = (W, E), k)$, we construct an election as follows. The set of candidates is

$$C := C_W \uplus C_E \uplus \{c, w\} \uplus D$$

with $C_W := \{c_u \mid u \in W\}$, $C_E := \{c_{uv} \mid \{u, v\} \in E\}$, and a set of dummy candidates D. In the following, the candidates in C_W and C_E are called *vertex candidates* and *edge candidates*, respectively. Furthermore, we construct the votes in a way such that w is the candidate that we would like to prevent from winning, c is the only candidate that can beat w, and D contains dummy candidates that can gain a score of at most one. In the multiset of votes V we have for every vertex $u \in W$ and for each incident edge $\{u, v\} \in E$ one vote of the type $c_u > c_{uv} > c > \dots$, that is, there are $2 \cdot |E|$ votes of this type, two for every edge. Additionally, V contains $|W| + k \cdot (k-1)$ votes in which w is at the first position and $|W| + 1$ votes in which c is at the first position. That is, the score of w exceeds the score of c by $k \cdot (k-1)$. In all votes, the remaining free positions between 2 and $k + \binom{k}{2} + 1$ are filled with dummies such that every dummy occurs in at most one vote at a position better than $k + \binom{k}{2} + 2$. This can be done using less than $|V| \cdot (k + \binom{k}{2} + 1)$ dummy candidates. In every vote the candidates that do not occur in this vote at a position less than $(k + \binom{k}{2} + 1)$ follow in arbitrary order.

Claim 2 *Graph G contains a clique K of size k iff candidate c can become plurality winner by deleting $k' := k + \binom{k}{2}$ candidates.*

"⇒": Delete the $k + \binom{k}{2}$ candidates that correspond to the vertices and edges of K. Then, for every of the $\binom{k}{2}$ deleted edge candidates we also deleted the two vertex candidates that correspond to the endpoints of the edge. Therefore, for every of the $\binom{k}{2}$ edges candidate c gets in the first position in two more votes. Hence, the score of candidate c is increased by $2 \cdot \binom{k}{2} = k \cdot (k-1)$ and the score of candidate w is not affected. Thus, the total score of w is $|W| + k \cdot (k-1)$ and the total score of c is $|W| + k \cdot (k-1) + 1$ and w is defeated by c.

"$\Leftarrow$": By construction, we cannot decrease the score of w and we cannot increase the score of a vertex candidate (which is at most $|W|-1$). Furthermore, by the deletion of at most k' candidates the score of a dummy candidate can become at most one, and the score of an edge candidate can become at most two. Hence, c is the only candidate that can prevent w from winning. Furthermore, as the deletion of at most k' dummies never moves c into a first position, we can assume that the solution deletes only edge and vertex candidates. Thus, it remains to that the only way to increase the score of c by at least $k \cdot (k-1)$ is to choose edge and vertex candidates that correspond to the vertices and edges of a clique of size k.

Let $C_{W'} \cup C_{E'}$ be a solution of size k', that is, deleting the candidates in $C_{W'} \cup C_{E'}$ prevents candidate c from winning. Let W' and E' be the corresponding vertices and edges and let $i := |W'|$. It is easy to see that $i \leq k$ since the deletion of an edge candidate moves c in exactly two votes from the third to the second position. Hence, in order to move c in at least $k \cdot (k-1)$ votes to the first position, we have to delete at least $(k \cdot (k-1))/2 = \binom{k}{2}$ edge candidates. Consequently, since $k' = \binom{k}{2} + k$, we can delete at most k further vertex candidates.

In the following, we show that we have to remove exactly k vertex candidates, that is, we must have $i = k$. Consider the election after deleting the candidates in $C_{W'}$. Let $E'_1 \subseteq E'$ be the set of edges with both endpoints in W' and let $E'_2 := E' \setminus E'_1$. Clearly, by deleting $C_{W'} \cup C_{E'}$ the score of c increases by at most $2 \cdot |E'_1| + |E'_2|$. Since $C_{W'} \cup C_{E'}$ is a solution, we obtain

$$2 \cdot |E'_1| + |E'_2| \geq \text{score}(c) - |W| - 1 \geq k \cdot (k-1). \tag{1}$$

Furthermore, we know that

$$|E'_1| + |E'_2| + i \leq k + \binom{k}{2}. \tag{2}$$

Inequality (1) implies that the score of c becomes maximum if E_1 is maximum, that is, if the graph (V', E'_1) is complete. In this case, E'_1 has cardinality $\binom{i}{2}$ and, hence, according to Inequality (2) the candidate set E_2 has cardinality at most $k' - \binom{i}{2} - i$ and the score of c is at most $2 \cdot \binom{i}{2} + k' - \binom{i}{2} - i + |W| + 1$. Assume that we have $i < k$, then $\text{score}(c) - |W| - 1 \leq 2 \cdot \binom{i}{2} + k' - \binom{i}{2} - i < k \cdot (k-1) = \text{score}(c) - |W| - 1$, a contradiction. Hence, we must have that $i = k$ and $|E'| = |E'_1| + |E'_2| = \binom{k}{2} = (k \cdot (k-1))/2$. Therefore, $E'_2 = \emptyset$ and (W', E') is a clique. □

5 Outlook

In this work, we investigated the parameterized complexity of four new digraph modification problems and of electoral candidate based control. Somewhat surprisingly, the problems turned out to be intractable in almost all settings. For instance, MAX-OUTDEGREE DELETION is W[2]-complete for two very restrictive digraph classes, tournaments and acyclic graphs.

We conclude this work with several concrete questions regarding future research.

- Recently, Erdélyi et al. [15] considered electoral control for "sincere-strategy preference-based approval voting" regarding its classical complexity. The parameterized complexity for meaningful parameterizations is still open.
- Regarding candidate control in plurality voting, we only gave W[1]-/W[2]-hardness results. The class containment was left open.
- We only considered the parameterized complexity for candidate control. There are many other settings to study, for example, control by adding or deleting votes.
- In Copeland$^\alpha$ voting with $0 < \alpha < 1$, the parameterized complexity with respect to the number of votes is open.
- In contrast to manipulation [10,29,31], for control up to now all investigations focused on worst-case scenarios. There seem to be no studies that are concerned with strategies that allow for efficient control in the average case or for "most" instances.

Acknowledgements

The authors are grateful to Rolf Niedermeier for the help with writing this paper and to the anonymous referees of *Theoretical Computer Science* for careful and constructive feedback. Furthermore, we like to thank Hannes Moser for introducing us to the method that we used for showing the class membership of the digraph problems in W[2].

References

[1] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications.* Springer, 2 edition, 2009. 13

[2] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6:227–241, 1989. 2

[3] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. How hard is it to control an election. *Mathematical and Computer Modelling*, 16(8-9):27–40, 1992. 2, 3, 8, 28, 29

[4] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. Fixed-parameter algorithms for Kemeny scores. In *Proc. of 4th AAIM '08*, volume 5034 of *LNCS*, pages 60–71. Springer, 2008. 2

[5] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. How similarity helps to efficiently compute Kemeny rankings. In *Proc. of 8th AAMAS '09*, 2009. To appear. 2

[6] N. Betzler, J. Guo, and R. Niedermeier. Parameterized computational complexity of Dodgson and Young elections. In *Proc. of 11th SWAT '08*, volume 5124 of *LNCS*, pages 402–413. Springer, 2008. 2

[7] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, and I. Schnoor. Approximability of manipulating an election. In *Proc. of 23rd AAAI '08*, pages 133–144. AAAI Press, 2008. 2

[8] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice (invited paper). In *Proc. of 33rd SOFSEM'07*, volume 4362 of *LNCS*, pages 51–69. Springer, 2007. 2

[9] R. Christian, M. R. Fellows, F. A. Rosamond, and A. M. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007. 2

[10] V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proc. of 21st AAAI '06*, pages 627–634, 2006. 33

[11] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):1–33, 2007. 2, 6, 7, 28

[12] R. G. Downey and M. R. Fellows. Threshold dominating sets and an improved version of W[2]. *Theoretical Computer Science*, 209:123–140, 1998. 20

[13] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999. 5, 8, 9, 13, 31

[14] G. Erdély, L. A. Hemaspaandra, J. Rothe, and H. Spakowski. On approximating optimal weighted lobbying, and frequency of correctness versus average-case polynomial time. In *Proc. of 16th FCT '07*, volume 4639 of *LNCS*, pages 300–311. Springer, 2007. 2

[15] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting broadly resists control. In *Proc. of 33rd MFCS '08*, volume 5162 of *LNCS*, pages 311–322. Springer, 2008. 2, 3, 33

[16] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of bribery in elections. In *Proc. of 21st AAAI '06*, pages 641–646. AAAI Press, 2006. 2

[17] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland voting broadly resist bribery and control. In *Proc. of 22nd AAAI'07*, pages 724–730, 2007. 2, 3, 4, 8, 22, 23, 28

[18] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Copeland voting fully resists constructive control. In *Proc. of 4th AAIM'08*, volume 5034, pages 165–176, 2008. 2, 3, 4, 7, 14, 23, 28

[19] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resists bribery and control. arXiv:0809.4484v2, 2008. 3, 8

[20] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*. Springer, 2009. To appear. 2

[21] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006. 5, 8

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

[23] G. Gutin and A. Yeo. Some parameterized problems on digraphs. *The Computer Journal*, 51(3):363–371, 2008. 6

[24] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171:255–285, 2007. 2, 3, 8, 28, 29, 31

[25] D. Jungnickel. *Graphs, Networks and Algorithms*. Springer, 3rd edition, 2008. 24

[26] D. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):311–330, 1953. 9

[27] R. Meir, A. Procaccia, J. S. Rosenschein, and A. Zohar. The complexity of stratigic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, 2008. 2

[28] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. 5, 8

[29] A. Procaccia and J. S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intellingence (JAIR)*, 28:157–181, 2007. 33

[30] L. Xia, V. Conitzer, A. D. Procaccia, and J. S. Rosenschein. Complexity of unweigthed coalitional manipulation under some common voting rules. In *Proc. of 21st IJCAI '09*, 2009. To appear. 2

[31] M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intellingence*, 173(2):392–412, 2009. 33