Towards a Dichotomy of Finding Possible Winners in Elections Based on Scoring Rules

Nadja Betzler[1,⋆] and Britta Dorn[2]

[1] Institut für Informatik, Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2, D-07743 Jena, Germany.
nadja.betzler@uni-jena.de
[2] Wilhelm-Schickard-Institut für Informatik, Universität Tübingen
Sand 13, D-72076 Tübingen, Germany.
bdorn@informatik.uni-tuebingen.de

Abstract. To make a joint decision, agents (or voters) are often required to provide their preferences as linear orders. To determine a winner, the given linear orders can be aggregated according to a voting protocol. However, in realistic settings, the voters may often only provide partial orders. This directly leads to the POSSIBLE WINNER problem that asks, given a set of partial votes, if a distinguished candidate can still become a winner. In this work, we consider the computational complexity of POSSIBLE WINNER for the broad class of voting protocols defined by scoring rules. A scoring rule provides a score value for every position which a candidate can have in a linear order. Prominent examples include plurality, k-approval, and Borda. Generalizing previous NP-hardness results for some special cases and providing new many-one reductions, we settle the computational complexity for all but one scoring rule. More precisely, for an unbounded number of candidates and unweighted voters, we show that POSSIBLE WINNER is NP-complete for all pure scoring rules except plurality, veto, and the scoring rule defined by the scoring vector $(2, 1, \ldots, 1, 0)$, while it is solvable in polynomial time for plurality and veto.

1 Introduction

Voting scenarios arise whenever the preferences of different parties (*voters*) have to be aggregated to form a joint decision. This is what happens in political elections, group decisions, web site rankings, or multiagent systems. Often, the voting process is executed in the following way: each voter provides his preference as a ranking (linear order) of all the possible alternatives (*candidates*). Given these rankings as an input, a *voting rule* produces a subset of the candidates (*winners*) as an output. However, in realistic settings, the voters may often only provide partial orders instead of linear ones: For example, it might be impossible for the voters to provide a complete preference list because the set of candidates is too large. In addition, not all voters might have given their preferences yet during the aggregation process, or new candidates might be introduced after some voters already have given their rankings. Moreover, one often has to

⋆ Supported by the DFG, Emmy Noether research group PIAF, NI 369/4 and PAWS, NI 369/10.

deal with partial votes due to incomparabilities: for some voters it might not be possible to compare two candidates or certain groups of candidates, be it because of lack of information or due to personal reasons. Hence, the study of partial voting profiles is natural and essential. One question that immediately comes to mind is whether any information on a possible outcome of the voting process can be given in the case of incomplete votes. More specifically, in this paper, we study the POSSIBLE WINNER problem: Given a partial order for each of the voters, can a distinguished candidate c win for at least one extension of the partial orders into linear ones?

Of course, the answer to this question depends on the voting rule that is used. In this work, we will stick to the broad class of voting protocols defined by *scoring rules*. A scoring rule provides a score value for every position that a candidate can take within a linear order. The scores of the candidates are then added up and the candidates with the highest score win. Many well-known voting protocols, including plurality, veto, and Borda, are realized by scoring rules. Other examples are the Formula 1 scoring, which uses the scoring rule defined by the vector $(10, 8, 6, 5, 4, 3, 2, 1, 0, ...)$, or k-approval, which is used in many political elections whenever the voters can express their preference for k candidates within the set of all candidates.

The POSSIBLE WINNER problem was introduced by Konczak and Lang [6] and has been further investigated since then for many types of voting systems [1, 7–10]. Note that the related NECESSARY WINNER problem can be solved in polynomial time for all scoring rules [10]. A prominent special case of POSSIBLE WINNER is MANIPULATION (see e.g. [2, 5, 11, 12]). Here, the given set of partial orders consists of two subsets; one subset contains linearly ordered votes and the other one completely unordered votes. Clearly, all NP-hardness results would carry over from MANIPULATION to POSSIBLE WINNER. However, whereas the case of weighted voters is settled by a full dichotomy [5] for MANIPULATION for scoring rules, we are not aware of any NP-hardness results for scoring rules in the unweighted voter case.

Let us briefly summarize known results for POSSIBLE WINNER for scoring rules. Correcting Konczak and Lang [6] who claimed polynomial-time solvability for all scoring rules, Xia and Conitzer [10] provided NP-completeness results for a class of scoring rules, more specifically, for all scoring rules that have four "equally decreasing score values" followed by another "strictly decreasing score value"; we will provide a more detailed discussion later. Betzler et al. [1] studied the parameterized complexity of POSSIBLE WINNER and, among other results obtained NP-hardness for k-approval in case of two partial orders. However, this NP-hardness result holds only if k is part of the input, and it does not carry over for fixed values of k. Further, due to the restriction to two partial votes, the construction is completely different from the constructions used in this work.

Until now, the computational complexity of POSSIBLE WINNER was still open for a large number of naturally appearing scoring rules. We mention k-approval for small values of k as an example: Assume that one may vote for a board that consists of five members by awarding one point each to five of the candidates (5-approval). Surprisingly, POSSIBLE WINNER turns out to be NP-hard even for 2-approval. Another example is given by voting systems in which each voter is allowed to specify a (small) group of favorites and a (small) group of most disliked candidates.

In this work, we settle the computational complexity of POSSIBLE WINNER for all pure scoring rules except the scoring rule defined by $(2, 1, \ldots, 1, 0)$.[1] For plurality and veto, we provide polynomial-time algorithms. The basic idea to show the NP-completeness for all remaining pure scoring rules can be described as follows. Every scoring vector of unbounded length must either have an unbounded number of positions with different score values or must have an unbounded number of positions with equal score values (or both). Hence, we give many-one reductions covering these two types and then combine them to work for all considered scoring rules. Scoring rules having an unbounded number of positions with different score values are treated in Section 2, where we generalize results from [10]. Scoring rules having an unbounded number of positions with equal score values are investigated in Section 3. Here, we consider two subcases. In one subcase, we consider scoring rules of the form $(\alpha_1, \alpha_2, \ldots, \alpha_2, 0)$, which we deal with in Section 3.2. In the other subcase, we consider all remaining scoring rules with an unbounded number of candidates (Section 3.1). Finally, we combine the obtained results in the main theorem (Section 4).

Preliminaries. Let $C = \{c_1, \ldots, c_m\}$ be the set of *candidates*. A *vote* is a linear order (i.e., a transitive, antisymmetric, and total relation) on C. An n-voter profile P on C consists of n votes $(v_1, \ldots, v_n)$ on C. A *voting rule* r is a function from the set of all profiles on C to the power set of C. *(Positional) scoring rules* are defined by scoring vectors $\overrightarrow{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ with integers $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$, the *score values*. More specifically, we define that a scoring rule r consists of a sequence of scoring vectors $s_1, s_2, \ldots$ such that for any $i \in \mathbb{N}$ there is a scoring vector for i candidates which can be computed in time polynomial in i.[2] Here, we restrict our results to *pure* scoring rules, that is for every i, the scoring vector for i candidates can be obtained from the scoring vector for $i-1$ candidates by inserting an additional score value at an arbitrary position (respecting the described monotonicity). This definition includes all of the common protocols like Borda or k-approval.[3] We further assume that $\alpha_m = 0$ and that there is no integer that divides all score values. This does not constitute a restriction since for every other voting system there must be an equivalent one that fulfills these constraints [5, Observation 2.2]. Moreover, we only consider *non-trivial* scoring rules, that is, scoring rules with $\alpha_1 \neq 0$ for a scoring vector of unbounded length.

For a vote $v \in P$ and a candidate $c \in C$, let the *score* $s(v, c) := \alpha_j$ where j is the position of c in v. For any profile $P = \{v_1, \ldots, v_n\}$, let $s(P, c) := \sum_{i=1}^{n} s(v_i, c)$. Whenever it is clear from the context which P we refer to, we will just write $s(c)$. The scoring rule will select all candidates c as winners for which $s(P, c)$ is maximized. Famous examples of scoring rules are Borda, that is, $(m-1, m-2, \ldots, 0)$, and k-approval, that is, $(1, \ldots, 1, 0, \ldots, 0)$ starting with k ones. Two relevant special cases of k-approval are plurality, that is $(1, 0, \ldots, 0)$, and veto, that is, $(1, \ldots, 1, 0)$.

---

[1] The class of pure voting rules still covers all of the common scoring rules. We only constitute some restrictions in the sense that for different numbers of candidates the corresponding scoring vectors can not be chosen completely independently (see Preliminaries).

[2] For scoring rules that are defined for any fixed number of candidates the considered problem can be decided in polynomial time, see [2, 9].

[3] Our results can also be extended to broad classes of "non-pure" (hybrid) scoring rules. Due to the lack of space, we defer the related considerations to the full version of this paper.

A *partial order* on C is a reflexive, transitive, and antisymmetric relation on C. We use $>$ to denote the relation given between candidates in a linear order and $\succ$ to denote the relation given between candidates in a partial order. Sometimes, we specify a whole subset of candidates in a partial order, e.g., $e \succ D$. This notation means that $e \succ d$ for all $d \in D$ and there is no specified order among the candidates in D. In contrast, writing $e > D$ in a linear order means that the candidates of D have an arbitrary but fixed order. A linear order v^l *extends* a partial order v^p if $v^p \subseteq v^l$, that is, for any $i, j \leq m$, from $c_i \succ c_j$ in v^p it follows that $c_i > c_j$ in v^l. Given a profile of partial orders $P = (v_1^p, \ldots, v_n^p)$ on C, a candidate $c \in C$ is a *possible winner* if there exists an extension $V = (v_1, \ldots, v_n)$ such that each v_i extends v_i^p and $c \in r(V)$. The corresponding decision problem is defined as follows.

POSSIBLE WINNER
Given: A set of candidates C, a profile of partial orders $P = (v_1^p, \ldots, v_n^p)$ on C, and a distinguished candidate $c \in C$.
Question: Is there an extension profile $V = (v_1, \ldots, v_n)$ such that each v_i extends v_i^p and $c \in r(V)$?

This definition allows that multiple candidates obtain the maximal score and we end up with a whole set of winners. If the possible winner c has to be unique, one speaks of a possible *unique winner*, and the corresponding decision problem is defined analogously. As discussed in the following paragraph, all our results hold for both cases.

In all many-one reductions given in this work, one constructs a partial profile P consisting of a set of linear orders V^l and another set of partial votes V^p. Typically, the positions of the distinguished candidate c are already determined in all votes from V^p, that is, $s(P, c)$ is fixed. The "interesting" part of the reductions is formed by the partial orders of V^p in combination with upper bounds for the scores of the non-distinguished candidates. For every candidate $c' \in C \backslash \{c\}$, the *maximum partial score* $s_p^{\max}(c')$ is the maximum number of points c' can make in V^p without beating c in P. The maximum partial scores can be adapted for the unique and for the winner case since beating c in the winner case just means that a candidate makes strictly more points than c and beating c in the unique winner case means that a candidate makes as least as many points as c. Since all reductions only rely on the maximum partial scores, all results hold for both cases. For all reductions given in this work, one can generate an appropriate set of linear votes that implement the required maximum partial scores for each candidate. A general construction scheme of these votes can be found in a long version of this work. For this paper, we will refer to this construction scheme as Construction 1.

Several of our NP-hardness proofs rely on reductions from the NP-complete EXACT 3-COVER (X3C) problem. Given a set of elements $E = \{e_1, \ldots, e_q\}$, a family of subsets $\mathcal{S} = \{S_1, \ldots, S_t\}$ with $|S_i| = 3$ and $S_i \subseteq E$ for $1 \leq i \leq t$, it asks whether there is a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that for every element $e_j \in E$ there is exactly one $S_i \in \mathcal{S}'$ with $e_j \in S_i$.

Due to the lack of space, several proofs and details had to be deferred to a full version of this paper.

2 An unbounded number of positions with different score values

Xia and Conitzer [10] showed that POSSIBLE WINNER is NP-complete for any scoring rule which contains four consecutive, equally decreasing score values, followed by another strictly decreasing score value. They gave reductions from X3C. Using some non-straightforward gadgetry, we extend their proof to work for scoring rules with an unbounded number of different, not necessarily equally decreasing score values.

We start by describing the basic idea given in [10] (using a slightly modified construction). Given an X3C-instance $(E, \mathcal{S})$, construct a partial profile $P := V^l \cup V^p$ on a set of candidates C where V^l denotes a set of linear orders and V^p a set of t partial orders. To describe the basic idea, we assume that there is an integer $b \geq 1$ such that $\alpha_b > \alpha_{b+1}$ and the difference between the score of the four following score values is equally decreasing, that is, $\alpha_b - \alpha_{b+1} = \alpha_{b+1} - \alpha_{b+2} = \cdots = \alpha_{b+3} - \alpha_{b+4}$, for a scoring vector of appropriate size. Then, $C := \{c, x, w\} \cup E' \cup B$ where $E' := \{e \mid e \in E\}$ and B contains $b - 1$ dummy candidates. The distinguished candidate is c. The candidates whose element counterparts belong to the set S_i are denoted by e_{i1}, e_{i2}, e_{i3}. For every $i \in \{1, \ldots, t\}$, the partial vote v_i^p is given by $B \succ x \succ e_{i1} \succ e_{i2} \succ e_{i3} \succ C'$, $B \succ w \succ C'$. Note that in v_i^p, the positions of all candidates except $w, x, e_{i1}, e_{i2}, e_{i3}$ are fixed. More precisely, w has to be inserted between positions b and $b+4$ maintaining the partial order $x \succ e_{i1} \succ e_{i2} \succ e_{i3}$. The maximum partial scores are set such that the following three conditions are fulfilled. First, regarding an *element candidate* $e \in E'$, inserting w behind e in two partial orders has the effect that e would beat c, whereas when w is inserted behind e in at most one partial order, c still beats e (Condition 1). Note that e may occur in several votes at different positions, e.g. e might be identical with e_{i1} and e_{j3} for $i \neq j$. However, due to the condition of "equally decreasing" scores, "shifting" e increases its score by the same value in all of the votes. Second, the partial score of x is set such that w can be inserted behind x at most $q/3$ times (Condition 2). Finally, we set $s_p^{\max}(w) = (t - q/3) \cdot \alpha_b + q/3 \cdot \alpha_{b+4}$. This implies that if w is inserted before x in $t - q/3$ votes, then it must be inserted at the last possible position, that is, position $b + 4$, in all remaining votes (Condition 3).

Now, having an exact 3-cover for $(E, \mathcal{S})$, it is easy to verify that setting w to position $b+4$ in the partial votes that correspond to the exact 3-cover and to position b in all remaining votes leads to an extension in which c wins. In a yes-instance for (C, P, c), it follows directly from Condition 1 and 2 that w must have the last possible position $b+4$ in exactly $q/3$ votes and position b in all remaining partial votes. Since $|E| = q$ and there are $q/3$ partial votes such that three element candidates are shifted in each of them, due to Condition 1, every element candidate must appear in exactly one of these votes. Hence, c is a possible winner in P if and only if there exists an exact 3-cover of E.

In the remainder of this section, we show how to extend the reduction to scoring rules with strictly, but not equally decreasing scoring values. The problem we encounter is the following: By sending candidate w to the last possible position in the partial vote v_i^p, each of the candidates e_{i1}, e_{i2}, e_{i3} improves by one position and therefore improves its score by the difference given between the corresponding positions. In [10], these differences all had the same value, but now, we have to deal with varying differences. Since the same candidate $e \in E'$ may appear in several votes at different positions, e.g. e might be identical with e_{i1} and e_{j3} for $i \neq j$, it is not clear how to set the maximum partial score of e. In order to cope with this situation, we add two copies $v_i^{p'}$ and $v_i^{p''}$ of every partial vote v_i^p, and permute the positions of candidates e_{i1}, e_{i2}, e_{i3} in these two copies such that each of them takes a different position in v_i^p, $v_i^{p'}$, $v_i^{p''}$. In this way, if the candidate w is sent to the last possible position in a partial vote and its two copies, each of the candidates e_{i1}, e_{i2}, e_{i3} improves its score by the same value (which is "added" to the maximum partial score). We only have to guarantee that whenever w is sent back in the partial vote v_i^p, then it has to be sent back in the two copies $v_i^{p'}$ and $v_i^{p''}$ as well. We describe how this can be realized using a gadget construction. More precisely, we give a gadget for pairs of partial votes. The case of three votes just uses this scheme for two pairs within the three partial votes.

Given a copy $v_i^{p'}$ for each partial vote v_i^p, we want to force w to take the last possible position in v_i^p if and only if w takes the last possible position in $v_i^{p'}$. We extend the set of candidates C by $2t$ additional candidates $D := \{d_1, \ldots, d_t, h_1, \ldots, h_t\}$. The set V^p consists of $2t$ partial votes $v_1^p, v_1^{p'}, \ldots, v_t^p, v_t^{p'}$ with
$v_i^p:\ B \succ x \succ S_i \succ d_1 \succ \cdots \succ d_i \succ h_{i+1} \succ \cdots \succ h_t \succ C_i',\ B \succ w \succ C_i'$
$v_i^{p'}: B \succ x \succ S_i \succ h_1 \succ \cdots \succ h_i \succ d_{i+1} \succ \cdots \succ d_t \succ C_i'', B \succ w \succ C_i''$,
for all $1 \leq i \leq t$, with C_i' and C_i'' containing the remaining candidates, respectively. Again, the maximum partial scores are defined such that w can be inserted behind x in at most $2q/3$ votes and must be inserted behind d_t or h_t in at least $2q/3$ votes. Further, for every candidate of D, the maximum partial score is set such that it can be shifted to a better position at most q times in a yes-instance. The candidate set E' is not relevant for the description of the gadget and thus we can assume that each candidate of E' can never beat c. We denote this construction as **Gadget 1**. Using some pigeonhole principle argument, one can show that Gadget 1 works correctly (proof omitted). Combining the construction of [10] with Gadget 1 and by using some further simple padding, one arrives at the following theorem.

Theorem 1. POSSIBLE WINNER *is NP-complete for a scoring rule if, for every positive integer x, there is a number m that is a polynomial function of x and, for the scoring vector of size m, it holds that* $|\{i \mid 1 \leq i \leq m-1 \text{ and } \alpha_i > \alpha_{i+1}\}| \geq x$.

3 An unbounded number of positions with equal score values

In the previous section, we showed NP-hardness for scoring rules with an unbounded number of different score values. In this section, we discuss scoring rules with an unbounded number of positions with equal score value. In the first subsection, we show NP-hardness for POSSIBLE WINNER for scoring rules with an unbounded number of "non-border" positions with the same score. That is, either before or after the group of equal positions, there must be at least two positions with a different score value. In the second subsection, we consider the special type that $\alpha_1 > \alpha_2 = \cdots = \alpha_{m-1} > 0$.

3.1 An unbounded number of non-border positions with equal score values

Here, we discuss scoring rules with non-border equal positions. For example, a scoring rules, such that, for every positive integer x, there is a scoring vector of size m such that there is an i, with $i < m-2$, and $\alpha_{i-x} = \alpha_i$. This property can be used to construct a basic "logical" tool used in the many-one reductions of this subsection: For two candidates c, c', having $c \succ c'$ in a vote implies that setting c such that it makes less than α_i points implies that also c' makes less than α_i points whereas all candidates placed in the range between $i-x$ and i make exactly α_i points. This can be used to model some implication of the type "$c \Rightarrow c'$" in a vote. For example, for $(m-2)$-approval this condition means that c only has the possibility to make zero points in a vote if also c' makes zero points in this vote whereas other all candidates make one point. Most of the reductions of this subsection are from the NP-complete MULTICOLORED CLIQUE (MC) problem [4]:

Given: An undirected graph $G = (X_1 \cup X_2 \cup \cdots \cup X_k, E)$ with $X_i \cap X_j = \emptyset$ for $1 \leq i < j \leq k$ and the vertices of X_i induce an independent set for $1 \leq i \leq k$.
Question: Is there a clique of size k?

Here, $1, \ldots, k$ are considered as different colors. Then, the problem is equivalent to ask for a *multicolored clique*, that is, a clique that contains one vertex for every color. To ease the presentation, for any $1 \leq i \neq j \leq k$, we interpret the vertices of X_i as red vertices and write $r \in X_i$, and the vertices of X_j as green vertices and write $g \in X_j$.

Reductions from MC are often used to show parameterized hardness results [4]. The general idea is to construct different types of gadgets. Here, the partial votes realize four kinds of gadgets. First, gadgets that choose a vertex of every color (vertex selection). Second, gadgets that choose an edge of every ordered pair of colors, for example, one edge from green to red and one edge from red to green (edge selection). Third, gadgets that check the consistency of two selected ordered edges, e.g. does the chosen red-green candidate refer to the same edge as the choice of the green-red candidate (edge-edge match)? At last, gadgets that check if all edges starting from the same color start from the same vertex (vertex-edge match).

We start by giving a reduction from MC that settles the NP-hardness of POSSIBLE WINNER for $(m-2)$-approval.

Lemma 1. POSSIBLE WINNER *is NP-hard for* $(m-2)$*-approval.*

Proof. Given an MC-instance $G = (X, E)$ with $X = X_1 \cup X_2 \cup \cdots \cup X_k$. Let $E(i,j)$ denote all edges from E between X_i and X_j. W.l.o.g. we can assume that there are integers s and t such that $|X_i| = s$ for $1 \leq i \leq k$, $|E(i,j)| = t$ for all i, j, and that k is odd. We construct a partial profile P on a set C of candidates such that a distinguished candidate $c \in C$ is a possible winner if and only if there is a size-k clique in G. The set of candidates $C := \{c\} \uplus C_X \uplus C_E \uplus D$, where $\uplus$ denotes the disjunctive union, is specified as follows:

- For $i \in \{1, \ldots, k\}$, let $C_X^i := \{r_1, \ldots, r_{k-1} \mid r \in X_i\}$ and $C_X := \bigcup_i C_X^i$.
- For $i, j \in \{1, \ldots, k\}, i \neq j$, let $C_{i,j} := \{rg \mid \{r, g\} \in E(i,j)\}$ and $C'_{i,j} := \{rg' \mid \{r, g\} \in E(i,j)\}$. Then, $C_E := (\bigcup_{i \neq j} C_{i,j}) \uplus (\bigcup_{i \neq j} C'_{i,j})$, i.e., for every edge $\{r, g\} \in E(i,j)$, the set C_E contains the four candidates rg, rg', gr, gr'.
- The set $D := D_X \uplus D_1 \uplus D_2$ is defined as follows. For $i \in \{1, \ldots, k\}$, $D_X^i := \{c_1^r, \ldots, c_{k-2}^r \mid r \in X_i\}$ and $D_X := \bigcup_i D_X^i$. For $i \in \{1, \ldots, k\}$, one has $D_1^i := \{d_1^i, \ldots, d_{k-2}^i\}$ and $D_1 := \bigcup_i D_1^i$. The set D_2 is defined as $D_2 := \{d^i \mid i = 1, \ldots, k\}$.

We refer to the candidates of C_X as *vertex-candidates*, to the candidates of C_E as *edge-candidates*, and to the vertices of D as *dummy-candidates*.

The partial profile P consists of a set of linear votes V^l and a set of partial votes V^p. In each extension of P, the distinguished candidate c gets one point in every partial vote (see definition below). Thus, by using Construction 1, we can set the maximum partial scores as follows. For every candidate $d^i \in D_2$, $s_p^{\max}(d^i) = |V^p| - s + 1$, that is, d^i must get zero points (*take a zero position*) in at least $s-1$ of the partial votes. For every remaining candidate $c' \in C\backslash(\{c\} \cup D_2)$, $s_p^{\max}(c') = |V^p| - 1$, that is, c' must get zero points in at least one of the partial votes.

In the following, we define $V^p := V_1 \cup V_2 \cup V_3 \cup V_4$. For all our gadgets only the last positions of the votes are relevant. Hence, in the partial votes it is sufficient to explicitly specify the "relevant candidates". More precisely, we define for all partial votes that each candidate that does not appear explicitly in the description of a partial vote is positioned before all candidates that appear in this vote.

The partial votes of V_1 realize the **edge selection gadgets**. Selecting an ordered edge (r, g) with $\{r, g\} \in E$ means to select the corresponding *pair of edge-candidates* rg and rg'. The candidate rg is used for the vertex-edge match check and rg' for the edge-edge match check. For every ordered color pair $(i, j), i \neq j$, V_1 has $t-1$ copies of the partial vote $\{rg \succ rg' \mid \{r, g\} \in E(i, j)\}$, that is, one of the partial votes has the constraint $rg \succ rg'$ for each $\{r, g\} \in E(i, j)$. The idea of this gadget is as follows. For every ordered color pair we have t edges and $t-1$ corresponding votes. Within one vote, one pair of edge-candidates can get the two available zero positions. Thus, it is possible to set all but one, namely the selected pair of edge-candidates to zero positions.

The partial votes of V_2 realize the **vertex selection gadgets**. Here, we need $k-1$ candidates corresponding to a selected vertex to do the vertex-edge match for all edges that are incident in a multicolored clique. To realize this, $V_2 := V_2^a \cup V_2^b$. In V_2^a we select a vertex and in V_2^b, by a cascading effect, we achieve that all $k-1$ candidates that correspond to this vertex are selected. In V_2^a, for every color i, we have $s-1$ copies of the partial vote $\{r_1 \succ c_1^r \mid r \in X_i\}$. In V_2^b, for every color i and for every vertex $r \in X_i$, we have the following $k-2$ votes.

For all *odd* $z \in \{1, \ldots, k-4\}$, $v_z^{r,i} : \{c_z^r \succ c_{z+1}^r, r_{z+1} \succ r_{z+2}\}$.
For all *even* $z \in \{2, \ldots, k-3\}$, $v_z^{r,i} : \{c_z^r \succ c_{z+1}^r, d_{z-1}^i \succ d_z^i\}$.
$v_{k-2}^{r,i} : \{c_{k-2}^r \succ d_{k-2}^i, r_{k-1}^i \succ d^i\}$.

The partial votes of V_3 realize the **vertex-edge match gadgets**. For $i, j \in \{1, \ldots, k\}$, for $j < i$, V_3 contains the vote $\{rg \succ r_j \mid \{r, g\} \in E, r \in X_i$, and $g \in X_j\}$ and, for $j > i$, V_3 contains the vote $\{rg \succ r_{j-1} \mid \{r, g\} \in E, r \in X_i$, and $g \in X_j\}$.

The partial votes of V_4 realize the **edge-edge match gadgets**. For every unordered color pair $\{i, j\}, i \neq j$ there is the partial vote $\{rg' \succ gr' \mid r \in X_i, g \in X_j\}$.

This completes the description of the partial profile. By counting, one can verify a property of the construction that is crucial to see the correctness: In total, the number of zero positions available in the partial votes is exactly equal to the sum of the minimum number of zero position the candidates of $C\backslash\{c\}$ must take such that c is a winner. We denote this property of the construction as *tightness*. It directly follows that if there is a candidate that takes more zero positions than desired, then c cannot win in this extension since then at least one zero position must be "missing" for another candidate.

$V_1:$		$\cdots > rg > rg'$	for $i, j \in \{1, \ldots, k\}, i \neq j, r \in X_i \backslash Q,$ and $g \in X_j \backslash Q$
$V_2^a:$		$\cdots > r_1 > c_1^r$	for $1 \leq i \leq k$ and $r \in X_i \backslash Q$
$V_2^b:$	$v_z^{r,i}$	$\cdots > r_{z+1} > r_{z+2}$	for $1 \leq i \leq k, r \in X_i \backslash Q$ for all $z \in \{1, 3, 5, \ldots, k-4\}$
	$v_z^{r,i}$	$\cdots > c_z > c_{z+1}$	for $1 \leq i \leq k, r \in X_i \backslash Q$ for all $z \in \{2, 4, 6, \ldots, k-3\}$
	$v_{k-2}^{r,i}$	$\cdots > r_{k-1} > d^i$	for $1 \leq i \leq k, r \in X_i \backslash Q$
	$v_z^{r,i}$	$\cdots > c_z^r > c_{z+1}^r$	for $1 \leq i \leq k, r \in X_i \cap Q$ for all $z \in \{1, 3, 5, \ldots, k-4\}$
	$v_z^{r,i}$	$\cdots > d_{z-1}^i > d_z^i$	for $1 \leq i \leq k, r \in X_i \cap Q$ for all $z \in \{2, 4, 6, \ldots, k-3\}$
	$v_{k-2}^{r,i}$	$\cdots > c_{k-2}^r > d_{k-2}^i$	for $1 \leq i \leq k, r \in X_i \cap Q$
$V_3:$		$\cdots > rg > r_j$	for $i, j \in \{1, \ldots, k\}, j < i, r \in X_i \cap Q,$ and $g \in X_j \cap Q$
		$\cdots > rg > r_{j-1}$	for $i, j \in \{1, \ldots, k\}, j > i, r \in X_i \cap Q,$ and $g \in X_j \cap Q$
$V_4:$		$\cdots > rg' > gr'$	for $i, j \in \{1, \ldots, k\}, i \neq j, r \in X_i \cap Q, g \in X_j \cap Q$

Fig. 1. Extension of the partial votes for the MC-instance. We highlight extensions in which candidates that do not correspond to the solution set Q take the zero positions.

Claim: The graph G has a clique of size k if and only if c is a possible winner in P.

"$\Rightarrow$" Given a multicolored clique Q of G of size k. Then, extend the partial profile P as given in Figure 1. One can verify that in the given extension every candidate takes the required number of zero positions.

"$\Leftarrow$" Given an extension of P in which c is a winner, we show that the "selected" candidates must correspond to a size-k clique. Recall that the number of zero positions that each candidate must take is "tight" in the sense that if one candidate gets an unnecessary zero position, then for another candidate there are not enough zero positions left.

First (edge selection), for $i, j \in \{1, \ldots, k\}, i \neq j$, we consider the candidates of $C_{i,j}$. The candidates of $C_{i,j}$ can take zero positions in one vote of V_3 and in $t-1$ votes of V_1. Since $|C_{i,j}| = t$ and in the considered votes at most one candidate of $C_{i,j}$ can take a zero position, every candidate of $C_{i,j}$ must take one zero position in one of these votes. We refer to a candidate that takes the zero position in V_3 as solution candidate rg_{sol}. For every non-solution candidate $rg \in C_{i,j} \backslash \{rg_{\text{sol}}\}$, its placement in V_1 also implies that rg' gets a zero position, whereas rg'_{sol} still needs to take one zero position (what is only possible in V_4).

Second, we consider the vertex selection gadgets. Here, analogously to the edge selection, for every color i, we can argue that in V_2^a, out of the set $\{r_1 \mid r \in X_i\}$, we have to set all but one candidate to a zero position. The corresponding *solution vertex* is denoted as r_{sol}. For every vertex $r \in X_i \backslash \{r_{\text{sol}}\}$, this implies that the corresponding dummy-candidate c_1^r also takes a zero position in V_2^a. Now, we show that in V_2^b we have to set all candidates that correspond to non-solution vertices to a zero position whereas all candidates corresponding to r_{sol} must appear only at one-positions. Since for every vertex $r \in X_i \backslash \{r_{\text{sol}}\}$, the vertex c_1^r has already a zero position in V_2^a, it cannot take a zero position within V_2^b anymore without violating the tightness. In contrast, for the selected solution candidate r_{sol}, the corresponding candidates $c_1^{r_{\text{sol}}}$ and r_{sol_1} still need to take one zero position. The only possibility for $c_1^{r_{\text{sol}}}$ to take a zero position is within vote $v_1^{r_{\text{sol}},i}$ by setting $c_1^{r_{\text{sol}}}$ and $c_2^{r_{\text{sol}}}$ to the last two positions. Thus, one cannot set r_{sol_2} and r_{sol_3} to a zero position within V_2. Hence, the only remaining possibility for r_{sol_2} and r_{sol_3} to get zero points remains within the corresponding votes in V_3. This implies for every non-solution vertex r that r_2 and r_3 cannot get zero points in V_3 and, thus, we have to choose to put them on zero positions in the vote $v_1^{r,i}$ from V_2^b. The same principle leads to a cascading effect in the following votes of V_2^b: One cannot choose to set the candidates $c_p^{r_{\text{sol}}}$ for $p \in \{1, \ldots, k-2\}$ to zero positions in votes of V_2^b with even index z and, thus, has to improve upon them in the votes with odd index z. This implies that all vertex-candidates belonging to r_{sol} only appear in one-positions within V_2^b and that all dummy candidates d_p^i for $p \in \{1, \ldots, k-2\}$ are set to one zero position. In contrast, for every non-solution vertex r, one has to set the candidates c_p^r, $p \in \{2, \ldots, k-2\}$, to zero positions in the votes with even index z, and, thus, in the votes with odd index z, one has to set all vertex-candidates belonging to r to zero positions. This further implies that for every non-solution vertex in the last vote of V_2^b one has to set d^i to a zero position and, since there are exactly $s-1$ non-solution vertices, d^i takes the required number of zero positions. Altogether, all vertex-candidates belonging to a solution vertex still need to be placed at a zero position in the remaining votes $V_3 \cup V_4$, whereas all dummy candidates of D and the candidates corresponding to the other vertices have already taken enough zero positions.

Third, consider the vertex-edge match realized in V_3. For $i, j \in \{1, \ldots, k\}, i \neq j$, there is only one remaining vote in which rg_{sol} with $r \in X_i$ and $g \in X_j$ can take a zero position. Hence, rg_{sol} must take this zero-position. This does imply that the corresponding incident vertex x is also set to a zero-position in this vote. If $x \neq r_{sol_i}$, then x has already a zero-position in V_2. Hence, this would contradict the tightness and rg_{sol} and the corresponding vertex must "match". Further, the construction ensures that each of the $k-1$ candidates corresponding to one vertex appears exactly in one vote of V_3 (for each of the $k-1$ candidates, the vote corresponds to edges from different colors). Hence, c can only be possible winner if a selected vertex matches with all selected incident edges.

Finally, we discuss the edge-edge match gadgets. In V_4, for $i, j \in \{1, \ldots, k\}, i \neq j$, one still need to set the solution candidates from $C_{i,j}$ to zero positions. We show that this can only be done if the two "opposite" selected edge-candidates match each other. For two such edges rg_{sol} and gr_{sol}, $r \in X_i, g \in X_j$, there is only one vote in V_4 in which they can get a zero position. If rg_{sol} and gr_{sol} refer to different edges, then in this vote only one of them can get zero points, and, thus, the other one still beats c. Altogether, if c is a possible winner, then the selected vertices and edges correspond to a multicolored clique of size k. □

By generalizing the reduction used for Lemma 1, one can show the following.

Theorem 2. POSSIBLE WINNER *is NP-complete for a scoring rule* r *if, for every positive integer* x*, there is a number* m *that is a polynomial function of* x *and, for the scoring vector of size* m*, there is an* $i \leq m-1$ *such that* $\alpha_{i-x} = \cdots = \alpha_{i-1} > \alpha_i$.

The following theorem is based on further extensions of the MC-reduction used to prove Lemma 1 and some additional reductions from X3C.

Theorem 3. POSSIBLE WINNER *is NP-complete for a scoring rule* r *if, for every positive integer* x*, there is a number* m *that is a polynomial function of* x *and, for the scoring vector of size* m*, there is an* $i \geq 2$ *such that* $\alpha_i > \alpha_{i+1} = \cdots = \alpha_{i+x}$.

3.2 Scoring rules of the form $(\alpha_1, \alpha_2, \ldots, \alpha_2, 0)$

The following theorem can be shown by using another type of reductions from X3C.

Theorem 4. POSSIBLE WINNER *is NP-complete for all scoring rules such that there is a constant* z *and for every* $m \geq z$*, the scoring vector of size* m *satisfies the conditions* $\alpha_1 > \alpha_2 = \alpha_{m-1} > \alpha_m = 0$ *and* $\alpha_1 \neq 2 \cdot \alpha_2$*.*

4 Main theorem

To state our main theorem, we still need the results for plurality and veto:

Proposition 1. POSSIBLE WINNER *can be solved in polynomial time for plurality and veto.*

The proof of Proposition 1 can be obtained by using some flow computations very similar to [1, Theorem 6].[4] Finally, we prove our main theorem.

Theorem 5. POSSIBLE WINNER *is NP-complete for all non-trivial pure scoring rules except plurality, veto, and scoring rules with size-*m*-scoring vector* $(2, 1, \ldots, 1, 0)$ *for every* $m \geq z$ *for a constant* z*. For plurality and veto it is solvable in polynomial time.*

Proof. (Sketch) Plurality and veto are polynomial-time solvable due to Proposition 1. Let r denote a scoring rule as specified in the first part of the theorem. Having any scoring vector different from $(1, 0, \ldots)$, $(1, \ldots, 1, 0)$, and $(2, 1, \ldots, 1, 0)$ for m candidates, it is not possible to obtain a scoring vector of one of these three types for $m' > m$ by inserting scoring values. Hence, since we only consider pure scoring rules, there must be a constant z such that r does not produce a scoring vector of type plurality, veto, or $(2, 1, \ldots, 1, 0)$ for all $m \geq z$. Now, we give a reduction from X3C (restricted to instances of size greater than z) to POSSIBLE WINNER for r that combines the reductions used to show Theorems 1 – 4. Let I with $|I| > z$ denote an X3C-instance. If there is a constant z' such that for all $m \geq z'$ the scoring vector corresponding to r is $(\alpha_1, \alpha_2, \ldots, \alpha_2, 0)$, then we can directly apply the reduction used to show Theorem 4. Otherwise, to make use of the MC-reductions, we apply the following strategy. Since EXACT 3-COVER and MULTICOLORED CLIQUE are NP-complete, there is a polynomial-time reduction from X3C to MC. Hence, let I' denote an MC-instance whose size is polynomial in $|I|$ and that is a yes-instance if and only if I is a yes-instance.

Basically, the problem we encounter by showing the NP-hardness for POSSIBLE WINNER for r by using one specific many-one reduction from the previous sections is that such a reduction produces a POSSIBLE WINNER-instance with a certain number m of candidates. Thus, according to the properties of the reductions described in the previous sections, one would need to ensure that the corresponding scoring vector of size m provides a sufficient number of positions with equal/different scores. This

[4] We also refer to Faliszewski [3] for further examples showing the usefulness of employing network flows for voting problems.

seems not to be possible in general. However, for every specific instance I or I', it is not hard to compute a (polynomial) number of positions with equal/different scores that is sufficient for I or I'. For example, for the X3C-reduction used to show Theorem 1, for an instance (E, S), it would be clearly sufficient to have $(|E| + |S|)^2$ positions with equal score since this is a trivial upper bound for the number of candidates used to "encode" (E, S) into an POSSIBLE WINNER-instance. Having computed a sufficient number for all types of reductions either for I or I' (details omitted), we can set x to be the maximum of all these numbers.

Then, we can make use of the following observation (proof omitted). For r, there is a scoring vector whose size is polynomial in x such that either $|\{i \mid \alpha_i > \alpha_{i+1}\}| \geq x$ or such that $\alpha_i = \cdots = \alpha_{i+x}$ for some i. Now, we can distinguish two cases. If $\alpha_i = \cdots = \alpha_{i+x}$ for some i, applying one of the reductions to I' or I given in Theorem 2 or Theorem 3 results in a POSSIBLE WINNER-instance that is a yes-instance (if and only if I' is a yes-instance and, thus, also) if and only if I is a yes-instance. Otherwise, we can apply the reductions given in the proof of Theorem 1 to I resulting in a POSSIBLE WINNER instance that is a yes-instance if and only if I is a yes-instance. Since the NP-membership is obvious, the main theorem follows. □

Acknowledgments. We thank Johannes Uhlmann, Rolf Niedermeier, and the referees of MFCS for constructive feedback that helped to improve this work.

References

1. N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proc. of 21st IJCAI '09*, 2009.
2. 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.
3. P. Faliszewski. Nonuniform bribery (short paper). In *Proc. 7th AAMAS '08*, pages 1569–1572, 2008.
4. M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *TCS*, 410(1):53–61, 2009.
5. E. Hemaspaandra and L. A. Hemaspaandra. Dichotomy for voting systems. *J. Comput. Syst. Sci.*, 73(1):73–83, 2007.
6. K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. of IJCAI-2005 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.
7. J. Lang, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Winner determination in sequential majority voting. In *Proc. of 20th IJCAI '07*, pages 1372–1377, 2007.
8. M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. In *Proc. of 20th IJCAI '07*, pages 1464–1469, 2007.
9. T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proc. of 22nd AAAI '07*, pages 3–8. AAAI Press, 2007.
10. L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proc. of 23rd AAAI '08*, pages 196–201. AAAI Press, 2008.
11. L. Xia, M. Zuckerman, A. D. Procaccia, V. Conitzer, and J. S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proc. 21st IJCAI '09*, 2009.
12. M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, 173(2):392–412, 2009.