A connection between sports and matroids: How many teams can we beat?

Ildikó Schlotter[1]* and Katarína Cechlárová[2]**

[1] Budapest University of Technology and Economics
H-1521 Budapest, Hungary
ildi@cs.bme.hu

[2] Institute of Mathematics, Faculty of Science, P.J. Šafárik University
Jesenná 5, 040 01 Košice, Slovakia
katarina.cechlarova@upjs.sk

Abstract. Given an on-going sports competition, with each team having a current score and some matches left to be played, we ask whether it is possible for our distinguished team t to obtain a final standing with at most r teams finishing before t. We study the computational complexity of this problem, addressing it both from the viewpoint of parameterized complexity and of approximation. We focus on a special case equivalent to finding a maximal induced subgraph of a given graph G that admits an orientation where the in-degree of each vertex is upper-bounded by a given function. We obtain a $\Theta(\log |V(G)|)$ approximation for this problem based on an asymptotically optimal approximation we present for a certain matroid problem in which we need to cover a base of a matroid by picking elements from a set family.

Keywords: sports elimination problem; graph labelling; parameterized complexity; approximation; matroids; gammoids.

1 Introduction

This paper was motivated by the so-called Sports Elimination problem: given a set of teams in a sports competition, each with a current score, and some matches to be played between the teams, can a distinguished team still win the competition? The study of this problem was initiated by a paper from Schwartz [29], and has been the subject of numerous papers [1, 3, 4, 14, 15, 18–20, 22, 24, 28, 30] since then. As the rules for score allocation in the competition influence the computational complexity of Sports Elimination, it has been studied for various kinds of competitions. For instance, Sports Elimination for baseball, where each match has a winner team that gets 1 point while the other team gets 0, can be solved in polynomial time [29]. By contrast, Sports Elimination for European football, where the winner gets 3 points, the loser

* Supported by the Hungarian Scientific Research Fund (OTKA grants no. K-108383 and no. K-108947).
** Supported by the grants VEGA 1/0344/14, VEGA 1/0142/15, and APVV-15-0091.

gets 0, and each team in a draw gets 1 point, turns out to be NP-hard [19]. Kern and Paulusma [20, 19] addressed the Generalized Sports Elimination(S) problem (gse(S) for short), where S describes the possible outcomes of a match, thus allowing the study of the Sports Elimination problem for different sports in a general framework.

In their paper, Kern and Paulusma [20] gave a dichotomy characterizing all sets S of possible outcomes for which gse(S) is polynomial-time solvable, and proving NP-hardness for the remaining cases. According to their results, the only easy cases are those equivalent (in some specific sense, as discussed later) to the case where S is *complete*, meaning that in each match k points are distributed between the two participating teams for some $k \in \mathbb{N}$, and all outcomes with the one team getting i points and the other team obtaining $k - i$ points for some $i \in \mathbb{N}$ are possible.

In this paper, we concentrate on the following generalization of gse(S) that we call MinStanding(S): given a set of teams in a competition, each with a current standing, and a set of remaining matches to be played, does there exist a way to play the remaining matches so that in the final standing at most r teams will have more points than our distinguished team t? In other words, can our team t finish not worse than at the $(r + 1)$-st place? Hoffman and Rivlin [18] showed that this question is hard to answer already for baseball, which has essentially the simplest score allocation rule.

To tackle the computational hardness of MinStanding(S), we applied the framework of parameterized complexity, investigated the possibilities for approximation, and searched for computationally tractable special cases. In particular, we mainly focused on the setting where S is complete, in which case we can decide the question for $r = 0$ in polynomial time, as implied by the results of Kern and Paulusma [19, 20].

Interestingly, MinStanding(S) where S is complete turns out to be equivalent to the following very natural graph problem: given an undirected graph G with vertex capacities $c : V(G) \to \mathbb{N}$ and an integer $r \in \mathbb{N}$, decide whether there exists a set of at most r vertices X in G such that $G - X$ admits an orientation where each vertex v has in-degree at most $c(v)$. We refer to this problem as Vertex Deletion to In-degree Bounded Orientation (vdibo), and we feel that it deserves to be studied on its own right. Indeed, the question whether a given graph with vertex capacities admits an in-degree bounded orientation is a well-studied problem in combinatorial optimization: Hakimi [16] gave a compact characterization for the existence of such orientations already in 1965. Since then, various extensions of the problem have been examined, see e.g. [5, 7, 10, 11]. Asahiro et al. [2] considered a variant of the degree-constrained orientation problem where a penalty function on the violated degree bounds is to be minimized, but to the best of our knowledge vdibo as defined here has not been studied before.

Further, we show that vdibo is strongly connected to the field of matroids. Namely, vdibo leads us to the following problem that we call Matroid Base Cover (mbc): given a matroid $\mathcal{M}$ and a family $\mathcal{F}$ of subsets of its universe, find a minimum number of sets from $\mathcal{F}$ whose union contains a base for $\mathcal{M}$. The fact that VDIBO reduces to the special case of MATROID BASE COVER where $\mathcal{M}$ is a gammoid and $\mathcal{F}$ is a partition of the universe of $\mathcal{M}$ allows us to apply a greedy approach for approximating VDIBO.

Our contribution. To handle the computational hardness of MINSTANDING(S), we mostly focus on approximation and fixed-parameter tractability.

For the case when the set S of possible outcomes is complete, we prove that MINSTANDING(S) is polynomially equivalent to the VERTEX DELETION TO IN-DEGREE BOUNDED ORIENTATION problem. To deal with VDIBO, we present a reduction to the MATROID BASE COVER problem. For this, in turn, we propose a greedy approximation algorithm in Theorem 8 that has logarithmic ratio: given an instance of MBC with optimum OPT, the algorithm of Theorem 8 yields a solution with at most $1 + \lfloor OPT \ln b \rfloor$ sets where b denotes the rank of the input matroid. Using our reduction from VDIBO to MBC, this algorithm immediately yields an approximation with $\theta(\log n)$ ratio for VDIBO (or, equivalently, for MINSTANDING(S)), where n is the number of vertices in the input graph (or the number of teams participating in the competition, in the context of MINSTANDING(S)); this result is stated in Theorem 7.

In Theorem 9 we also prove that for any $\varepsilon > 0$, approximating MBC within a ratio of $(1 - \varepsilon) \ln |Y|$ is NP-hard, where Y is the universe of the input matroid. This shows that the ratio of our greedy approximation proposed for MBC is tight. Furthermore, in Theorem 10 we also show that the approximation ratio of the algorithm of Theorem 7 for VDIBO is roughly optimal, since for any $\varepsilon > 0$, VDIBO is NP-hard to approximate with ratio $\varepsilon \ln n$.

Regarding the general case where the set S of outcomes can be arbitrary[3], we prove various intractability results (parameterized hardness results, NP-hardness for certain restricted cases and strong inapproximability), and also identify polynomial-time solvable special cases. Namely, in Theorem 2 we show that MINSTANDING(S) is W[2]-hard with parameter r; hence, the problem remains hard even if the number r of teams allowed to beat our distinguished team can be assumed to be small. Similarly, Theorem 3 proves W[1]-hardness for the dual parameter "number of teams to beat" (i.e., $n - r$). Notably, both these results hold also if the graph underlying the input has restricted structure: Theorem 2 holds even for bipartite graphs, and Theorem 3 remains true for $K_{1,4}$-free graphs. To contrast these strong intractability results, we propose a simple cubic-time algorithm for MINSTANDING(S) in the special case where the graph G underlying the input is a tree. Using this algorithm, we prove polynomial-time solvability also for the case where G is a forest (Corollary 1) or has maximum degree 2 (Corollary 2). In Theorem 5 we further prove that Corollary 2 is sharp in the sense that MINSTANDING(S) remains NP-hard if each vertex in G has degree 2 or 3 (or even if all vertices have degree 3); hence we obtain a clear limit of tractability of the problem with respect to vertex degrees in the underlying graph. Finally,

[3] Actually we need certain small technical assumptions on S, essentialy to rule out the degenerate case when the matches can only have one possible outcome.

in Theorem 6 we prove that MinStanding(S) for general S is essentially inapproximable with *any* approximation ratio, even if we allow for a large additive term as well.

Organization. The next section contains the preliminaries used in the paper, and provides our model for the MinStanding(S) problem. In Section 3 we consider MinStanding(S) for general S, and present our classical and parameterized hardness results, as well as our polynomial-time algorithms for trees, forests, and graphs with maximum degree 2. Section 4 deals with MinStanding(S) in the case where S is complete. After showing its equivalence with vdibo, we give the reduction from vdibo to mbc, the greedy approximation to mbc (and its application to vdibo), and to close the section, the contrasting inapproximability results. We sum up our work and give conclusions in Section 5.

2 Preliminaries

2.1 Formulation of the model

Following the works [20, 4], we regard each match in a given competition to be played by two teams: the 'home' team and the 'away' team. Each match can end in several ways, and these possible outcomes are described by pairs of the form (α, β), where α and β denote the number of points obtained by the home and the away team, respectively. We denote the set of all possible outcomes by S. For instance, $S = \{(1,0),(0,1)\}$ describes baseball, while $S = \{(3,0),(1,1),(0,3)\}$ describes the score allocation in European football competitions. Notice that we allow for an asymmetry between the home and the away team in a match.

Kern and Paulusma [20] proved that we can restrict our attention to so-called *normalized* sets of outcomes, where $S = \{(\alpha_0, \beta_0), \ldots, (\alpha_k, \beta_k)\}$ with $k \geq 1$ and

$$\alpha_0 = 0,\ \alpha_1 = 1 < \alpha_2 < \cdots < \alpha_k \text{ and } \beta_0 > \beta_1 > \cdots > \beta_{k-1} \geq 1,\ \beta_k = 0. \quad (1)$$

Although they considered the Generalized Sports Elimination(S) problem, their observations hold also in the context of MinStanding(S). Hence, throughout the paper we assume all sets of outcomes to be normalized. In most of our hardness proofs though, we will only make use of the more relaxed property that S contains some pairs $(\alpha, 0)$ and $(0, \beta)$ for some positive α and β. We call a set S with this property *well-based*, and we call $(\alpha, 0)$ and $(0, \beta)$ the *extreme outcomes* in S. Another special property that we will focus on is the case where S is *complete*, meaning that $S = \{(i, k-i) \mid i = 0, 1, \ldots, k\}$ for some $k \in \mathbb{N}$.

Given some normalized set S of outcomes, it is easy to see that we can assume w.l.o.g. that the distinguished team t has no remaining matches to play, as otherwise we can just assume that t wins as many points in the remaining matches as possible (as in this case all other teams gain as few points from these matches as possible, since S is normalized). Hence, we can calculate the total score s_t that our team t must have at the end of the competition. Then, our question is reduced to asking whether all the remaining matches can be played in a way such that in the end at most r teams have score higher than s_t.

We can represent this situation using a directed multigraph, where vertices correspond to teams participating in the competition (except for our distinguished team t), and arcs correspond to the remaining matches to be played in the sense that each arc (t_1, t_2) represents a match where t_1 is the home team and t_2 is the away team. The set of outcomes for a match are represented by a set of labels on the corresponding arc; thus, choosing an outcome for a match can be thought of as choosing a label for the given arc. Furthermore, each vertex is assigned an integer *capacity* whose value is the number of points the corresponding team can gain during the remainder of the competition (in addition to its already achieved score) without overtaking our distinguished team t, that is, without ending up with a score higher than s_t. We can formalize our question as the following graph labelling problem.

MinStanding(S):
Instance: A triple (G, c, r) where $G = (V, A)$ is a directed multigraph, $c : V \to \mathbb{R}$ describes vertex capacities, and r is an integer.
Question: Does there exist an assignment $p : A \to \{0, \ldots, k\}$ such that the number of vertices in V violating the inequality

$$scr_p(v) := \sum_{a=(v,u)\in A} \alpha_{p(a)} + \sum_{a=(u,v)\in A} \beta_{p(a)} \leq c(v) \tag{2}$$

is at most r?

Vertices violating Inequality (2) according to some score assignment are called *violating* vertices. A score assignment under which there are at most r violating vertices, is called *r-violating.*

2.2 Parameterized complexity

In the study of computationally hard problems, parameterized complexity plays an important role. In this framework, we associate with each input I of a given problem Q an integer ℓ called the *parameter*, and we examine how this parameter affects the complexity of the problem. Namely, we consider the running time of an algorithm as a function of both the input size $|I|$ and the parameter ℓ, allowing us a more refined view on the computational complexity of Q.

We say that a parameterized problem Q is *fixed-parameter tractable*, if it admits an algorithm that runs in time $f(\ell)|I|^{O(1)}$ for some computable function f. Note that in the running time, the dependence on $|I|$ has constant degree, independent of ℓ. Therefore, such algorithms can be efficient in practice for small values of the parameter, even if the function f is, say, exponential.

To argue that a problem is not fixed-parameter tractable, one can show its hardness by means of a *parameterized* or *FPT-reduction.* Given two parameterized problems Q and Q', an FPT-reduction from Q to Q' is a function f such that (1) f can be computed by a fixed-parameter tractable algorithm, (2) for each input (x, ℓ) of Q it holds that $(x, \ell) \in Q$ if and only if $f(x, \ell) = (x', \ell') \in Q'$, and (3) $\ell' \leq g(\ell)$ for some function g.

The basic class of parameterized intractability is $\mathsf{W}[1]$; if a parameterized problem is $\mathsf{W}[1]$-hard, then this yields strong evidence that it does not admit an FPT algorithm. The class $\mathsf{W}[2]$ is the next class in the parameterized class hierarchy

$$\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{W}[P],$$

where each containment is believed to be strict. A problem is called $\mathsf{W}[1]$-hard (respectively, $\mathsf{W}[2]$-hard), if all problems in $\mathsf{W}[1]$ (respectively, $\mathsf{W}[2]$) can be reduced to it by an FPT-reduction.

For more details on parameterized complexity, we refer the reader to the recent monograph by Downey and Fellows [8].

2.3 Graph theory and matroids

Graphs. Throughout the paper we will use standard notation concerning graphs, see e.g. [6]. For a directed graph $G = (V, A)$, we will denote by $\delta^-(v)$ and $\delta^+(v)$ the *in-degree* and the *out-degree* of a vertex $v \in V$, respectively. For a subgraph H of G, we let $\delta^-_H(v)$ denote the in-degree of v restricted to H, that is, the number of arcs in H that enter v. Similarly, we $\delta^+_H(v)$ denote the number of arcs in H that leave v. By the *degree* $\delta(v)$ of v we mean the sum of its in- and out-degree, so $\delta(v) = \delta^-(v) + \delta^+(v)$. For an undirected graph $G = (V, E)$, we denote the *degree* of a vertex $v \in V$ by $d(x)$.

For a set X of arcs, edges, or vertices in a (directed or undirected) graph G, we write $G - X$ to denote the subgraph of G obtained from G by deleting X; when deleting vertices, all incident edges are deleted too. Also, for a subgraph H of G, we let $G - H$ be the graph $G - V(H)$.

Matroids. Given a set U and a family $\mathcal{I} \subseteq 2^U$ of subsets of U, we say that $\mathcal{M} = (\mathcal{I}, U)$ is a *matroid* over the *universe* U, if (i) $\emptyset \in \mathcal{I}$, (ii) if $X \in \mathcal{I}$ and $Y \subseteq X$, then $Y \in \mathcal{I}$, and (iii) if $X, Y \in \mathcal{I}$ and $|X| < |Y|$, then there exists some $y \in Y \setminus X$ for which $X \cup \{y\} \in \mathcal{I}$ as well. The subsets of U contained in $\mathcal{I}$ are called *independent sets*. A *base* of the matroid $\mathcal{M}$ is a maximal independent set in $\mathcal{I}$. From the definitions it follows that any two bases of $\mathcal{M}$ have the same size; this number is called the *rank* of $\mathcal{M}$. Analogously, the *rank* $r(W)$ of a set $W \subseteq U$ is defined as the maximum size of any independent set contained in W (hence, the rank of $\mathcal{M}$ equals to the rank of U).

If every subset of U that has size at most $r(U)$ is an independent set of $\mathcal{M}$, then $\mathcal{M}$ is called a *uniform matroid* of rank $r(U)$. Given a directed graph $G = (V, A)$ with two sets of terminals $T_1, T_2 \subseteq V$, we can define a matroid over the universe T_1 in which some set $W \subseteq T_1$ is independent if there exist $|W|$ vertex-disjoint paths leading in G from W to distinct vertices of T_2; a matroid defined this way is called a *gammoid*. Given two matroids $\mathcal{M}_1$ and $\mathcal{M}_2$ defined over the universes U_1 and U_2, respectively, the *direct sum* of $\mathcal{M}_1$ and $\mathcal{M}_2$ is a matroid whose universe is the disjoint union of U_1 and U_2, and in which some set is independent if it can be obtained as the union of an independent set from $\mathcal{M}_1$ and an independent set from $\mathcal{M}_2$.

For more background on matroids, see e.g. [25] or a less technical texbook [21] that explains the main ideas of how classical combinatorial optimization problems have been generalized to the notion of a matroid.

3 MinStanding(S) for general S

The special case of MinStanding(S) where we fix the integer r to be zero yields exactly the gse(S) problem. Thus, the complexity dichotomy given in [20] has the following consequences.

Theorem 1 (Kern & Paulusma [20]). *Given a normalized set S of outcomes that is not complete, the restriction of* MinStanding(S) *to the case $r = 0$ is* NP*-complete. By contrast, if S is complete, then* MinStanding(S) *on instances where $r = 0$ can be solved in polynomial time.*

For general values of r, strong NP-completeness of MinStanding(S) has been proved in [18] for the special case $S = \{(1, 0), (0, 1)\}$. The following theorem shows that MinStanding(S) remains intractable even if the number r of teams that are allowed to beat our distinguished team (in other words, the number of violating vertices) is small, and the set of outcomes is the simplest possible.

Theorem 2. MinStanding(S) *is strongly* NP*-complete, and also* W[2]*-hard with parameter r for any well-based set S of outcomes, even if the (undirected version of the) input graph is bipartite.*

Proof. We reduce from the Red-Blue Dominating Set problem which is NP-hard and also W[2]-hard [9]. The input of this parameterized problem consists of a bipartite graph $G = (R, B; E)$, with the two color classes R and B referred to as red and blue, and an integer parameter r. The task is to decide if there exists a set of at most r red vertices that dominate all blue vertices. We present a polynomial-time reduction from this problem to MinStanding(S) that is an FPT-reduction as well, proving both NP-hardness and W[2]-hardness; containment in NP is trivial.

Let $G = (R, B; E)$ and r be our input instance of Red-Blue Dominating Set. Let $(0, \beta)$ and $(\alpha, 0)$ be extreme outcomes in S. We construct an instance (G', c, r) of MinStanding(S) as follows. To define G', we simply orient the edges of G such that the vertices of R become sources. Then we set the vertex capacities as follows.

$$c(v) = \begin{cases} 0, & \text{if } v \in R, \\ (\delta(v) - 1)\beta, & \text{if } v \in B. \end{cases}$$

First, let $D \subseteq R$ be a set of r red vertices dominating all blue ones in G. We can define a score assignment on the arcs of G' in which teams corresponding to vertices of D win all their remaining matches gaining α points in each of them, while teams corresponding to vertices of $R \setminus D$ lose all their remaining matches gaining 0 point in each match. In such an assignment p, $scr_p(v) = 0$ for each $v \in R \setminus D$, and since each blue vertex has at least one neighbor in D, we also have $scr_p(v) \leq (\delta(v)-1)\beta$ for each $v \in B$. Thus, we have a score assignment for G' in which the only violating vertices are the r vertices in D.

For the other direction, suppose that we are given an r-violating score assignment p. If there exists a blue vertex v that is violating in p, then we modify p as follows: we let v gain β points on all but one of its incoming arcs, and we let v gain 0 point on the remaining (arbitrarily chosen) arc leading from some red vertex u to v. Note that our blue vertex v ceases to be violating in the resulting score assignment p', and the only vertex that might become newly violating is the red vertex u. Hence, p' is also r-violating, and contains less violating blue vertices than p does. Repeating this procedure as long as it is possible, we arrive at an r-violating score assignment where the set D of violating vertices is a subset of R; recall $|D| \leq r$.

Since any blue vertex $v \in B$ receives β points from each arc leading from $R \setminus D$ to v, the capacity of v ensures that v must have a neighbor in D, as otherwise it would gain $\delta(v)\beta > c(v)$ points, contradicting to the fact that v cannot be violating. Hence, D must indeed dominate all blue vertices. $\square$

The result of Theorem 2 shows that even for small values of r, it is hard to decide whether our team can finish no worse than at the $(r+1)$-st place. Similarly, we can ask whether our team can beat at least a few other teams. In other words, we aim for a score assignment where at least a few number of vertices are non-violating. NP-hardness of this problem is already implied by Theorem 2, and below we prove intractability of this situation in the parameterized sense as well. Namely, we show that MinStanding(S) is W[1]-hard with respect to the parameter $n-r$, where n is the number of vertices in the input graph. The fact that neither r, nor its dual parameter $n-r$ leads to fixed-parameter tractability (as opposed to many examples in the parameterized literature) suggests that MinStanding(S) is a considerably tough problem.

Theorem 3. MinStanding(S) *is* W[1]*-hard with parameter* $|V(G)|-r$ *for any well-based set* S *of outcomes, even if the (undirected version of the) input graph* G *is* $K_{1,4}$*-free.*[4]

Proof. We give a simple FPT-reduction from the W[1]-hard Independent Set problem, which is known to be W[1]-hard even on $K_{1,4}$-free graphs [17]. Let G be the input graph and ℓ the parameter given. The constructed instance of MinStanding(S) will be $(\overrightarrow{G}, c, |V(G)| - \ell)$ where $\overrightarrow{G}$ is an arbitrarily oriented version of G, and c is the constant zero function.

[4] In the version of this paper published in *Algorithmica*, we erronously claimed this result to hold for *claw-free* graphs instead of $K_{1,4}$-free graphs; we later corrected this mistake in an erratum. We are grateful for Matthias Mnich who kindly pointed out this error.

Now, it is easy to see that a set X of vertices in G is independent if and only if there is a score assignment on $\overrightarrow{G}$ in which vertices of X are not violating. (Again, we make use of the fact that S is well-based.) $\square$

On the positive side, we were able to establish that MinStanding(S) can be solved in polynomial time on trees. Though sports competitions where a team's success is measured by its score collected over several matches do not usually use a tree as the underlying structure for scheduling matches, the result of Theorem 4 can still be applicable in practice: after most of the matches have already been played, a situation might arise when the structure of the remaining matches does not contain any cycles (see also Corollary 1).[5]

Theorem 4. *For any well-based set S of outcomes,* MinStanding(S) *can be solved in $O(r^2n)$ (and hence also in $O(n^3)$) time, if the input graph is a directed tree on n vertices.*

Proof. Let (G, c, r) be our input; we may assume $r \leq |V(G)|$. We perform dynamic programming, so we will assume that $T = \bar{G}$ is a rooted tree with root v_{root}. For any vertex $v \in V(T)$, we let $C_v \subseteq V(T)$ denote the children of v, T_v the subtree of T rooted at v, and $G_v = G[T_v]$.

We compute the following values for each $v \in V(T)$. First, for each $i \in \{0, \dots, r\}$ we let $scr^i(v)$ be the minimum score of v in any i-violating score assignment for G_v where v is not violating. If no such assignment exists, then we set $scr^i(v) = +\infty$. Second, we also compute the non-negative integer $r^*(v)$ defined as the minimum i such that there is a score assignment for G_v in which i vertices of $V(T_v) \setminus \{v\}$ are violating; note that here we allow v to be violating, and thus such a score assignment may be altogether $(i+1)$-violating.

Observe that ($\star$) there is an i-violating score assignment on G_v for some $i \in \{0, \dots, r\}$ exactly if $scr^i(v) \leq c(v)$ or $r^*(v) + 1 \leq i$. Hence, (G, c, r) is a yes-instance if and only if $scr^r(v_{root}) \leq c(v_{root})$ or $r^*(v_{root}) + 1 \leq r$.

Clearly, $r^*(v) = 0$ and $scr^i(v) = 0$ for each leaf v of T and $0 \leq i \leq r$.

For a non-leaf node v, we have

$$r^*(v) = \sum_{x \in C_v} \min\{i \mid scr^i(x) \leq c(x) \text{ or } r^*(x) + 1 \leq i\}. \tag{3}$$

To see this, notice that if we allow v to be a violating vertex, then to each arc vx (or xv) incident to v in G_v we can assign the outcome where the child x of v gains 0 points from this arc. Hence, the minimum number of violating vertices in such an assignment, not counting v itself, can be computed by summing up the minimum values i for each $x \in C_v$ such that G_x admits an i-violating score assignment; by ($\star$) this yields exactly the formula above.

To compute the values $scr^i(v)$ for a non-leaf node v, $i \in \{0, \dots, r\}$, we first compute for each $x \in C_v$ and each $j \in \{0, \dots, r\}$ the minimum gain of v from

[5] In fact, this situation occurs at latest when each team has at most one remaining match to be played.

the arc vx (or xv) in any score assignment for $G[T_x \cup \{v\}]$ that is j-violating on the vertices of T_x; we denote this value by $g_v(x, j)$. It is easy to check that we have the following (in the formula below, q can take values from $\{0, \dots, k\}$, and we use the convention that taking the minimum over an empty set yields $+\infty$)

$$g_v(x,j) = \begin{cases} 0 & \text{if } r^*(x) < j, \\ \min\{\alpha_q \mid scr^j(x) + \beta_q \leq c(x)\} & \text{if } r^*(x) \geq j \text{ and } vx \in A(G), \\ \min\{\beta_q \mid scr^j(x) + \alpha_q \leq c(x)\} & \text{if } r^*(x) \geq j \text{ and } xv \in A(G). \end{cases} \tag{4}$$

Let $C_v = \{x_1, \dots, x_t\}$ be the children of v. An i-violating assignment p on G_v can be viewed as the union of certain score assignments on the subgraphs $G[T_{x_h} \cup \{v\}]$ for $h = 1, \dots, t$. Clearly, if v is not violating in p, then the number of violating vertices in these score assignments sums up to at most i, because p is i-violating. The minimum possible score that v gains when taking the union of such score assignments is, by the definitions, exactly

$$g_v^{min}(i) = \min\left\{\sum_{h=1}^{t} g_v(x_h, j_h) \mid j_1, \dots j_t \in \mathbb{N} \text{ with } \sum_{h=1}^{t} j_h = i\right\}. \tag{5}$$

From this we immediately obtain that

$$scr^i(v) = \begin{cases} g_v^{min}(i) & \text{if } g_v^{min}(i) \leq c(v), \\ +\infty & \text{otherwise.} \end{cases} \tag{6}$$

Having determined a recurrence for the values we aim to compute, let us now look at the time needed for their calculation. We perform the necessary computations in a bottom-up manner, so when computing the values $r^*(v)$ and $scr^i(v)$, $i \in \{0, \dots, r\}$, we assume that the corresponding values are already available for each child x of v. Then $r^*(v)$ can clearly be computed using (3) in $O(r|C_v|)$ time. The computation of all the values $g_v(x, j)$ using (4) can be done in total time $O(kr|C_v|)$.

However, computing $g_v^{min}(i)$ for some i using the recurrence (5) directly would take exponential time, so we need a slightly more elaborate approach. Namely, we can apply dynamic programming: instead of taking into account all the t children of v at once, we deal with them in t steps, considering only the children $x_1, \dots, x_\ell$ in the ℓ-th step. To this end, for each $\ell = 1, \dots, t$ we let

$$g_v^{min}(x_1 \to x_\ell, i) = \min\left\{\sum_{h=1}^{\ell} g_v(x_h, j_h) \mid j_1, \dots j_\ell \in \mathbb{N} \text{ with } \sum_{h=1}^{\ell} j_h = i\right\}.$$

Note that $g_v^{min}(x_1 \to x_1, i) = g_v(x, i)$, and for $\ell \geq 2$ we get

$$g_v^{min}(x_1 \to x_\ell, i) = \min\left\{g_v^{min}(x_1 \to x_{\ell-1}, j) + g(x_\ell, j') \mid j, j' \in \mathbb{N} \text{ with } j + j' = i\right\}. \tag{7}$$

One computation of the form (7) takes $O(r)$ time, so calculating $g_v^{min}(x_1 \to x_\ell, i)$ for each $\ell = 1, \dots, t$ and $i = 0, \dots, r$ takes $O(r^2 t) = O(r^2|C_v|)$ time. By $g_v^{min}(i) = g_v^{min}(x_1 \to x_t, i)$ and (6), from this we immediately obtain also the values $scr^i(v)$, $i \in \{0, \ldots, r\}$. Hence, the overall running time of the algorithm is

$$\sum_{v \in V(T)} |C_v|(O(r) + O(kr) + O(r^2)) = |V(G)|O(kr + r^2) = O(|V(G)|r^2).$$

□

Given an instance I of MinStanding(S) where the underlying graph G is a directed forest, we can solve I by processing each connected component of G independently. Supposing that G consists of the trees $T_1, \ldots, T_c$, for each $i \in \{1, \ldots, c\}$ we can determine the minimum number of violating vertices v_i for each tree T_i; the minimum number of violating vertices in any score assignment for I is then simply $\sum_{i=1}^{c} v_i$. This yields the following corollary of Theorem 4.

Corollary 1. *For any well-based set S of outcomes,* MinStanding(S) *can be solved in $O(r^2 n)$ (and hence also in $O(n^3)$) time if the input graph is a directed forest on n vertices.*

A further consequence of Theorem 4 is that MinStanding(S) can also be solved in polynomial time if the input graph has maximum degree at most 2. This means that if each team in the competition has at most two remaining matches to be played, then our problem becomes tractable.

Corollary 2. *For any well-based set S of outcomes,* MinStanding(S) *can be solved in $O(r^2 n)$ (and hence also in $O(n^3)$) time, if the input graph has n vertices and maximum degree 2.*

Corollary 2 relies on the observation that an undirected graph with maximum degree at most 2 is a collection of disjoint cycles and paths. Paths can be handled by Theorem 4 directly. To calculate the minimum number of violating vertices for a cycle, we need to pick an arbitrary arc of the cycle, try assigning each of the $|S| = k + 1$ possible outcomes to it, and then apply the algorithm of Theorem 4 on each of the resulting instances. For details of this method, we refer to Cechlárová et al. [4, Theorems 3 and 4] who apply the same arguments for the Sports Elimination problem.

Remark 1. We remark that Theorem 4 and hence Corollaries 1 and 2 hold even if S is not necessarily well-based, by the polynomial-time reduction provided by Kern and Paulusma [20] that produces an equivalent instance with a normalized (and thus, well-based) set of outcomes.

We now present a result sharply contrasting Corollary 2 by proving that MinStanding(S) is NP-complete if each vertex of the input graph has degree either 2 or 3, or even if all vertices have degree 3. Since many sports competitions schedule matches in a way that at any moment every team has roughly the same number of remaining matches, Theorem 5 has great practical relevance. Together with Corollary 2, it shows that deciding whether our distinguished team is still able to obtain a good final standing is computationally intractable while there are teams with three remaining matches, but becomes easily solvable in the moment when all teams have at most two matches to play. Theorem 5 also shows that, somewhat surprisingly, MinStanding(S) remains NP-complete even if each team has the same number of remaining matches. Let us remark that the case when $r = 0$ (that is, the Sports Elimination problem) was already known to be NP-complete for any S that is *not* complete, even if each team has at most 3 remaining matches [4]; Theorem 5 proves a similar result for general values of r that holds for practically *any* set S of outcomes (it only assumes S to be well-based).

Theorem 5. MinStanding(S) *is strongly* NP*-complete for any well-based set S of outcomes, even if the (undirected version of the) input graph is bipartite and (a) each vertex has degree* 2 *or* 3*, or (b) each vertex has degree* 3*.*

Proof. We present a reduction from the following NP-hard [12] variant of Exact Cover by 3-Sets which we denote by 3X3C. The input of 3X3C consists of a set U with $3n$ elements for some $n \in \mathbb{N}$, together with a family $\mathcal{T}$ of triplets of elements of U, such that each element of U is contained in at least two but at most three triplets. The task is to decide whether there exist n triplets in $\mathcal{T}$ whose union is U; such a family of triplets is called an *exact cover* for U.

Observe that 3X3C can be turned into an equivalent instance of Red-Blue Dominating Set: let $G = (\mathcal{T}, U; E)$ be the bipartite graph where we set $\mathcal{T}$ as the set of red vertices and U as the set of blue vertices, and we connect some $T \in \mathcal{T}$ with some $u \in U$ if and only if $u \in T$. Observe $\mathcal{T}' \subseteq \mathcal{T}$ is a set of at most n red vertices dominating all blue ones if and only if $\mathcal{T}'$ is an exact cover for U. Hence, the instance (G, n) of Red-Blue Dominating Set is equivalent to our input instance of 3X3C.

Thus, we can re-use the reduction from Red-Blue Dominating Set to MinStanding(S) presented in the proof of Theorem 2. It remains to observe that each vertex of G has degree either 2 or 3, and since the reduction used in the proof of Theorem 2 only orients the edges of the given graph (without altering vertex degrees), the constructed instance (G', c, n) of MinStanding(S) has the properties required in case (a).

To obtain the result for case (b), that is, when all vertices have degree 3, it suffices to use the same reduction from the variant of 3X3C where each element appears in *exactly* three triplets; this problem is also NP-hard [13]. □

To examine the possibilities of approximation for MinStanding(S), we define its optimization variant in the standard way: instead of giving the integer r as part of an instance for MinStanding(S), the aim is to determine the minimum value for r that results in a yes-instance. Thus, in the optimization variant of MinStanding(S), each input instance is a pair (G, c) formed by a multigraph $G = (V, A)$ and a capacity function $c : V \to \mathbb{R}$. We say that a set of vertices $R \subseteq V$ is a *solution* for (G, c) if there exists a score assignment for G in which all violating vertices (with respect to c) are contained in R. The task is then to find a solution of minimum size.

Regarding the (in)approximability of MinStanding(S) for the cases where S is not complete, the NP-hardness of MinStanding(S) for $r = 0$ implies that MinStanding(S) cannot be approximated within *any* multiplicative factor, since deciding whether the optimum is zero or not already is NP-hard. However, this reasoning does not rule out the existence of an approximation algorithm that has an additive term in the size of its output, such as the algorithm of Theorem 7. Below we show that allowing for an additive term in the approximation guarantee does not yield an opportunity for an efficient approximation algorithm.

Theorem 6. *If* $\mathsf{P} \neq \mathsf{NP}$*,* S *is a set of outcomes that is not complete, and* ε *is a constant with* $0 < \varepsilon < 1$*, then no polynomial-time algorithm exists that for a given instance* I *of* MinStanding(S) *outputs a solution for* I *with size at most* $\alpha(I)OPT + n^{1-\varepsilon}$*, where* n *is the number of vertices in the input graph,* OPT *is the optimum solution size for* I*, and* α *is an arbitrary function.*

Proof. Suppose for contradiction that such an algorithm $\mathcal{A}$ exists. Using $\mathcal{A}$, we are going to give an algorithm for the special case of MinStanding(S) with $r = 0$ that runs in polynomial time. By the results of Kern & Paulusma [20] (as repeated in Theorem 1), this problem is NP-hard for any S that is not complete.

We define a constant c_0 as the integer $\lfloor \frac{1-\varepsilon}{\varepsilon} \rfloor + 1$. Given some instance $I = (G, c, 0)$ of MinStanding(S), we construct an instance I' for MinStanding(S) by taking n^{c_0} disjoint copies of G and setting the capacity function to equal c on each of the copies. Note that the number of vertices in I' is $N = n^{c_0+1}$, which is polynomial in n. We run $\mathcal{A}$ on I'.

Clearly, if I is a yes-instance, then the optimum solution size OPT for I' is 0, since an assignment with no violating vertices exists for (G, c) and hence for any number of disjoint copies of it as well. Thus, algorithm $\mathcal{A}$ is guaranteed to output a solution of size at most $\alpha(I')OPT + N^{1-\varepsilon} = n^{(c_0+1)(1-\varepsilon)}$.

By contrast, if I is a no-instance, then any solution R for I' must contain at least one vertex from each copy of G, implying $|R| \geq n^{c_0}$. In particular, the output of $\mathcal{A}$ must have size at least n^{c_0}. By $c_0 > \frac{1-\varepsilon}{\varepsilon}$ we quickly get that

$$n^{(c_0+1)(1-\varepsilon)} < n^{c_0},$$

which implies that we can distinguish between yes- and no-instances for MinStanding(S) with $r = 0$. $\square$

4 MinStanding(S) for complete S

From now on, let us deal with the special case of MinStanding(S) where $S = \{(i, k-i) \mid 0 \leq i \leq k\}$ for some $k \in \mathbb{N}$.

It is not hard to see that an instance (G, c, r) of MinStanding(S) is equivalent with an instance (G', c, r) of MinStanding($\{(1,0),(0,1)\}$), where G' is the digraph obtained from G by replacing each arc ab with k parallel arcs from a to b. Indeed, assigning the outcome $(i, k-i) \in S$ on some arc ab in G corresponds precisely to assigning the outcome $(1,0)$ to i arcs and the outcome $(0,1)$ to $k-i$ arcs among the k parallel arcs leading from a to b in G'.

Considering the case $S = \{(1,0),(0,1)\}$, we can reformulate our problem as follows.

> **Vertex Deletion to In-degree Bounded Orientation** (VDIBO):
> *Instance:* A triple (G,c,r) where $G=(V,E)$ is an undirected multigraph, $c: V \to \mathbb{N}$ represents degree bounds, and r is an integer.
> *Question:* Does there exist a set R of at most r vertices such that there exists an orientation of $G-R$ in which $\delta^-(v) \leq c(v)$ for each $v \in V \setminus R$?

Similarly as for MinStanding(S), we call a set R of vertices fulfilling the above requirement a *solution*, and we say that R is *minimal* if no proper subset of R is a solution.

By Theorem 2, we know that VDIBO is strongly NP-complete, and even W[2]-hard with parameter r. Nevertheless, MinStanding(S) for a complete S, and therefore VDIBO as well, is in XP with respect to the parameter r: the problem can be solved in polynomial time for each fixed value of r by the simple brute force algorithm checking for all possible sets R of size r whether R is a solution for the given instance: deleting R from the input graph and solving the remainder (setting also $r=0$ to forbid additional deletions) can be done in polynomial time by Theorem 1.

Here we provide a greedy approximation algorithm for VDIBO with approximation ratio $O(\ln |V(G)|)$. In Section 4.1, we reduce our problem instance to a certain matroid covering problem, for which we propose an approximation algorithm in Section 4.2.

4.1 Reduction to matroid base cover

Suppose we are given an instance (G,c,r) of VDIBO. Let us start from an arbitrary orientation of G. We call a vertex v *unsaturated*, *exact*, or *oversaturated*, if $\delta^-(v)$ is smaller, equal to, or larger than $c(v)$, respectively. We will call the number $exc(v) = \delta^-(v) - c(v)$ the *excess* of v.

Preprocessing phase. We apply the following simple step as long as it is possible: if there is a directed path leading from an unsaturated vertex to an oversaturated vertex, then reverse all arcs on this paths. When there are several such paths, we can choose one arbitrarily (even though the graph obtained at the end of this phase depends on our choices).

Observe that reversing a path leading from a to b during the preprocessing phase decreases the excess of b, while it does not turn a into an oversaturated vertex. After the preprocessing phase, let $\overrightarrow{G}$ be the obtained digraph, and U the set of all vertices in $\overrightarrow{G}$ which are reachable from some unsaturated vertex. Observe that U contains all unsaturated vertices but contains no oversaturated vertex, and moreover, no arcs leave U in $\overrightarrow{G}$. This implies that any minimal solution R must be disjoint from U. Hence, we can delete U from our graph without changing the solvability of our instance; let D_1 be the digraph $\overrightarrow{G} - U$.

Note that each vertex of D_1 is either oversaturated or exact, and thus has non-negative excess.

The following series of lemmas help us to reformulate our question in terms of matroids.

Lemma 1. *A set $R \subseteq V(D_1)$ is a solution for (G, c, r) if and only if there exists a set $\mathcal{P}$ of arc-disjoint paths in D_1 such that each path of $\mathcal{P}$ leads from R to an oversaturated vertex of D_1, and for each vertex $v \in V(D_1) \setminus R$, exactly $exc(v)$ paths of $\mathcal{P}$ end in v.*

Proof. By the discussion above, a set $R \subseteq V(D_1)$ is a solution for (G, c, r) if and only if it is a solution for $(G-U, c_{|G-U}, r)$ where $c_{|G-U}$ is the restriction of c to the vertices of $G-U$ (meaning $c_{|G-U}(v) = c(v)$ for each $v \in V(G) \setminus U$).

Suppose that R is a set of vertices and $\mathcal{P}$ a set of paths fulfilling the conditions of the lemma. We claim that reversing all paths of $\mathcal{P}$ in D_1 yields an orientation where each vertex not in R has in-degree at most its capacity. Observe that the only vertices whose in-degree is affected by the reversal of a directed path P are the endvertices of P. If $P \in \mathcal{P}$, then P starts at a vertex of R, so reversing P decreases the in-degree of its endpoint by one, but does not modify the indegree of any other vertex in $V(D_1) \setminus R$. Since for each $v \in V(D_1) \setminus R$, exactly $exc(v)$ paths of $\mathcal{P}$ end in v, the claim follows. Therefore, R indeed yields a solution.

For the other direction, suppose that some set $R \subseteq V(D_1)$ is a solution for (G, c, r), and thus for $(G-U, c_{|G-U}, r)$. Let D_R be an orientation of $G-U$ where each vertex v not in R has in-degree at most $c(v)$. Let Δ be the subgraph of D_1 spanned by those arcs in D_1 which are oriented in the opposite direction in D_R as in D_1 (recall that both digraphs were obtained by orienting $G-U$). Notice that for each vertex $v \in V(D_1) \setminus R$, the in-degree of v in D_R is exactly $\delta^-_{D_1}(v) - \delta^-_{\Delta}(v) + \delta^+_{\Delta}(v)$, from which we know that

$$\delta^-_{\Delta}(v) - \delta^+_{\Delta}(v) \geq exc_{D_1}(v) \geq 0. \tag{8}$$

We show that Δ contains a set of paths as reqired by the lemma.

Let $\mathcal{P}$ be an inclusion-wise maximal set of arc-disjoint paths in Δ, each starting at some vertex of R, such that $|\mathcal{P}_v| \leq exc(v)$ for each $v \in V(D_1) \setminus R$ where $\mathcal{P}_v$ is the set of paths in $\mathcal{P}$ ending at v. For contradiction, suppose that there exists at least one vertex $w \in V(D_1) \setminus R$ with $|\mathcal{P}_w| < exc(w)$. Let W be a maximal walk ending in $\Delta - \mathcal{P}$ that ends at w and uses each arc at most once, and let x be its starting vertex. If $x \in R$, then we can turn W into a path from x to w by shortcutting all previously visited nodes in W, which contradicts the maximality of $\mathcal{P}$. Thus, $x \notin R$, and so $|\mathcal{P}_x| \leq exc(x)$ which implies

$$\delta^-_{\Delta-\mathcal{P}}(x) - \delta^+_{\Delta-\mathcal{P}}(x) = \delta^-_{\Delta}(x) - \delta^+_{\Delta}(x) - |\mathcal{P}_x| \geq \delta^-_{\Delta}(x) - \delta^+_{\Delta}(x) - exc(x) \geq 0,$$

where the last inequality follows from (8). However, the maximality of W shows that there is no arc of $\Delta - \mathcal{P}$ entering x not already used by W, yielding $\delta^-_{\Delta-\mathcal{P}}(x) - \delta^+_{\Delta-\mathcal{P}}(x) < 0$, a contradiction. This finishes the proof of the lemma. $\square$

Observe that using Lemma 1 we can easily decide if a given set R of vertices is a solution, by computing a maximum flow in an appropriate network. However, to construct such an R we need some additional tools.

Let D_2 be the graph obtained from D_1 as follows. For each vertex $v \in V(D_1)$, we add a set $X(v)$ of $exc(v)$ new vertices together with the arcs $\{vx \mid x \in X(v)\}$. We let $X = \bigcup_{v \in V(D_1)} X(v)$. The following lemma is a direct consequence of Lemma 1.

Lemma 2. *A set $R \subseteq V(D_1)$ is a solution for (G, c, r) if and only if there exists a set of $|X|$ arc-disjoint paths in D_2 leading from R to distinct vertices of X.*

Next, we proceed from arc-disjoint paths to vertex-disjoint ones, by constructing the following digraph D_3. First, for each $v \in V(D_2) \setminus X$ we replace v with a set $Y(v) = \{v_{\to u} \mid vu \in A(D_2)\}$ of new vertices. Intuitively, $v_{\to u}$ will host those paths running through v in D_2 that leave v towards u. We let $Y = \bigcup_{v \in V(D_2) \setminus X} Y(v)$, so the vertex set of D_3 will be the disjoint union of X and Y. Second, for each arc $vu \in A(D_2)$ with $u \in X$ we replace vu with the arc $v_{\to u}u$, and for each arc $vu \in A(D_2)$ with $u \notin X$ we replace vu with the arcs $\{v_{\to u}y \mid y \in Y(u)\}$.

Lemma 3. *For any set $R \subseteq V(D_1)$, there exist $|X|$ arc-disjoint paths in D_2 leading from R to distinct vertices of X if and only if there exist $|X|$ vertex-disjoint paths in D_3 leading from a subset of $\bigcup_{v \in R} Y(v)$ to X.*

Let D_4 be the digraph obtained by reversing each arc of D_3. We say that a subset $T \subseteq Y$ is *linked* to X, if there are $|T|$ vertex-disjoint paths leading from X to T in D_4. It is well known that the family of those vertex sets T which are linked to X in D_4 form the independent sets of a matroid $\mathcal{M}_X$ defined over Y; in particular, $\mathcal{M}_X$ is a gammoid [26] (see also [25]). Hence, by Lemmas 2 and 3, we can formulate our task as follows. Given the matroid $\mathcal{M}_X$ and the partition $\mathcal{Y} = \{Y(v) \mid v \in V(D_1)\}$ of its universe Y, we need to decide if there exists at most r blocks from the partition $\mathcal{Y}$ whose union contains a base of $\mathcal{M}_X$. In other words, we aim for a base of $\mathcal{M}_X$ that can be *covered* with the smallest number of blocks from $\mathcal{Y}$.

To deal with the above described problem, we actually examine a somewhat more general case. Namely, we define the following problem that we name Matroid Base Cover.

> **Matroid Base Cover** (MBC):
> *Instance:* A matroid $\mathcal{M}$ defined over a universe Y, and a family $\mathcal{F} \subseteq 2^Y$ of subsets of Y.
> *Task:* Find a minimum number of sets from $\mathcal{F}$ whose union covers a base of $\mathcal{M}$.

In Theorem 8 of Section 4.2 we propose a polynomial-time approximation algorithm for MBC that produces a solution with $1 + \lfloor OPT \ln b \rfloor$ sets, where OPT is the number of sets in an optimal solution and b is the rank of the given matroid.

Applying this algorithm on the matroid $\mathcal{M}_X$ and the partition $\mathcal{Y}$ defined above, we obtain an approximation algorithm for VDIBO. Notice that the rank b of $\mathcal{M}_X$ is $|X| = \sum_{v \in V(D_1)} exc(v)$, and therefore we must have $b \leq |V(G)|^2$ if our instance is solvable. This, together with the arguments of this section, yields the following theorem.

Theorem 7. *There is a polynomial-time approximation algorithm for* VDIBO *that produces a solution of size at most*

$$1 + \left\lfloor OPT \ln \left(\sum_{v \in V(D_1)} exc(v) \right) \right\rfloor \leq 1 + \lfloor OPT \cdot 2 \ln |V(G)| \rfloor ,$$

where G is the input graph, D_1 is the digraph obtained after the preprocessing step, and OPT is the size of an optimal solution.

4.2 Greedy approximation for Matroid Base Cover

Here we give a greedy approximation algorithm for MBC. Let $\mathcal{M}$ be our matroid given, with a family $\mathcal{F}$ of subsets of its universe Y.

We are going to construct a base B of $\mathcal{M}$. At the i-th step, we will obtain an independent set I_i as follows. We start from $I_0 = \emptyset$, and the set I_i is constructed from I_{i-1} by choosing a set $F_i \in \mathcal{F}$ that maximizes $r(F_i \cup I_{i-1})$. We let $I_i \supset I_{i-1}$ be a maximal independent set in $F_i \cup I_{i-1}$. The algorithm terminates if I_i is a base of $\mathcal{M}$; we let i^* denote the index of this step. We let b the rank of $\mathcal{M}$, that is, $b = |I_{i^*}|$.

Suppose that B_{OPT} is a base that can be covered with the minimum number OPT of sets from $\mathcal{F}$. Let $\mathcal{F}_{OPT}$ denote the family of these sets. The key observation in the analysis of our algorithm is the following.

Lemma 4. *For each $i = 1, \ldots, i^*$ we have*

$$|I_i| - |I_{i-1}| \geq \frac{b - |I_{i-1}|}{OPT}. \tag{9}$$

Proof. Let us consider the i-th step, and let $F_1, F_2, \ldots, F_{i-1}$ be the sets of $\mathcal{F}$ chosen by the previous steps; we denote the family of these sets by $\mathcal{F}_{i-1}$. We say that a set $F \in \mathcal{F}$ has *value* $r(F \cup I_{i-1}) - |I_{i-1}|$. Our greedy algorithm picks the set in $\mathcal{F}$ with the highest value.

Adding the sets in $\mathcal{F}_{OPT}$ to $\mathcal{F}_{i-1}$ yields a family of at most $i - 1 + OPT$ sets whose union contains B_{OPT}. By properties of a matroid, this means that summing up the values of the sets in $\mathcal{F}_{OPT}$ yields at least $b - |I_{i-1}|$. Hence, at least one of the sets from $\mathcal{F}_{OPT}$ has value at least $(b - |I_{i-1}|)/OPT$. As our greedy algorithm picks the set F_i having the highest value in $\mathcal{F}$, we know that this value is at least $(b - |I_{i-1}|)/OPT$. Therefore, we obtain $|I_i| - |I_{i-1}| = r(F_i \cup I_{i-1}) - |I_{i-1}| \geq (b - |I_{i-1}|)/OPT$, as promised. □

Applying (9) repeatedly, we obtain

$$|I_i| \geq \frac{b}{OPT}\left(1+\left(1-\frac{1}{OPT}\right)+\left(1-\frac{1}{OPT}\right)^2+\cdots+\left(1-\frac{1}{OPT}\right)^{i-1}\right)$$
$$= b\left(1-\left(1-\frac{1}{OPT}\right)^i\right).$$

Clearly, the algorithm terminates if $|F_i| > b-1$. This leads us to the fact that i^* is the smallest integer such that

$$\left(1-\frac{1}{OPT}\right)^i < \frac{1}{b},$$

that is,

$$i < \frac{\ln b}{OPT \ln \frac{OPT}{OPT-1}} OPT.$$

Considering the function $f(x) = x \ln \frac{x}{x-1}$, it is not hard to see that $f(x) > 1$ for any $x > 1$. Therefore, we obtain that the number of blocks chosen by the greedy algorithm can be upper bounded by $i^* \leq 1 + \lfloor OPT \ln b \rfloor$. Thus, we have the following theorem.

Theorem 8. *There is a polynomial-time approximation algorithm for* Matroid Base Cover *that produces a solution containing at most* $1 + \lfloor OPT \ln b \rfloor$ *sets, where* b *is the rank of the matroid given and* OPT *is the number of sets in an optimal solution.*

4.3 Inapproximability results

In Theorem 8 presented in the previous section, we proposed an algorithm for MBC that always yields a solution of size $1 + \lfloor OPT \ln b \rfloor$ where b is the rank of the input matroid and OPT is the size of an optimum solution. To complement this result, the following theorem shows that this is essentially optimal, since no polynomial-time algorithm can provide an approximation with ratio $(1-\varepsilon)|Y|$ unless $\mathsf{P} = \mathsf{NP}$, where Y is the universe of the input matroid; note that $b \leq |Y|$ always holds.

Theorem 9. *For any* $\varepsilon > 0$*, it is* NP*-hard to approximate* Matroid Base Cover *within a ratio of* $(1-\varepsilon)\ln|Y|$*, where* Y *is the universe of the input matroid* $\mathcal{M}$*, even if*

- $\mathcal{M}$ *is the direct sum of uniform matroids each with rank* 1*, and*
- *the set family* $\mathcal{F} \subseteq 2^Y$ *forms a partition of* Y*.*

Proof. We present a reduction from Set Cover. In this problem, we are given a set $U = \{u_1, \ldots, u_n\}$ together with a family of subsets $\mathcal{S} \subseteq 2^U$, and the task is to find a minimum number of sets from $\mathcal{S}$ whose union contains U. In a recent paper, Moshkovitz [23] proved that Set Cover on inputs of size N cannot be approximated within a ratio of $(1-\varepsilon)\ln N$ for any $\varepsilon > 0$ in polynomial time, unless $\mathsf{P} = \mathsf{NP}$.

We are going to construct an instance of Matroid Base Cover with input matroid $\mathcal{M}$ and set family $\mathcal{F}$. Suppose that the sets given for Set Cover are $\mathcal{S} = \{S_1, \dots, S_m\}$. We define the universe of matroid $\mathcal{M}$ as $Y = \{y_{i,j} \mid u_i \in S_j, 1 \le i \le n, 1 \le j \le m\}$. Note that $|Y|$ equals the size N of the Set Cover instance. We let $\mathcal{M}$ be the direct sum of the matroids M_i, $1 \le i \le n$, where M_i is the uniform matroid with universe $Y_i = \{y_{i,j} \mid u_i \in S_j, 1 \le j \le m\}$ and rank 1. We define $\mathcal{F}$ as the partition of Y whose blocks are the sets $F_j = \{y_{i,j} \mid u_i \in S_j, 1 \le i \le n\}$ for $j = 1, \dots, m$.

Clearly, the sets in $\mathcal{F}$ bijectively correspond to the sets in $\mathcal{S}$, and some $\mathcal{F}' \subseteq \mathcal{F}$ covers a base of $\mathcal{M}$ if and only if it contains at least one element from each of the sets Y_i, which happens exactly if the corresponding sets $\mathcal{S}' \subseteq \mathcal{S}$ form a covering of set U. Hence, applying an approximation algorithm with ratio $(1-\varepsilon)\ln|Y|$ for some $\varepsilon > 0$ to the constructed instance of Matroid Base Cover we obtain an approximation algorithm for Set Cover with ratio $(1-\varepsilon)\ln N$. As the contruction can be performed in polynomial time, the theorem follows. $\square$

Remark 2. It is well known that matroids obtained as the direct sum of uniform matroids with rank 1 belong to the class of gammoids, and also to the class of graphic matroids (see e.g. [25]). Hence, the inapproximability result of Theorem 9 also holds for these special cases as well.

Let us also mention here a result by Richey and Punnen [27], who proved that given a graph G and a partition $\mathcal{E}$ of its edges, finding a minimum number of sets in $\mathcal{E}$ whose union contains a spanning tree of G is NP-hard[6]. Since this problem is in fact the special of Matroid Base Cover where $\mathcal{M}$ is graphic and $\mathcal{F}$ is a partition of the universe of $\mathcal{M}$, Theorem 9 can be thought of as a generalization of the result by Richey and Punnen.

Recall that vdibo can be reduced to the special case of Matroid Base Cover where the matroid $\mathcal{M}$ given is a gammoid, and the family $\mathcal{F}$ of subsets (from which we have to cover a base) is a partition of the universe. By Theorem 9 and Remark 2, the approximation algorithm for Matroid Base Cover provided by Theorem 8 is essentially optimal. However, this does not rule out the possibility of approximating vdibo directly with a better approximation ratio. To address this issue, we prove below that the approximation algorithm presented in Theorem 7 has roughly optimal ratio: assuming $\mathsf{P} \neq \mathsf{NP}$, no polynomial-time algorithm can provide an approximation with ratio $\varepsilon \ln n$ for any constant ε with n denoting the number of vertices in the input graph; this contrasts the bound $1 + \lfloor OPT \cdot 2\ln n \rfloor$ on the solution size guaranteed by the algorithm of Theorem 7.

Theorem 10. *For any constant $\varepsilon > 0$, it is NP-hard to approximate* vdibo *with ratio $\varepsilon \ln n$, where n is the number of vertices in the input graph.*

[6] In fact, Richey and Punnen [27] proved NP-hardness for a slightly more general version of this problem, but their reduction proves the stronger result stated here.

Proof. Again, we present a reduction from Set Cover. Let us be given an instance I_{SC} of Set Cover with ground set $U = \{u_1, \dots, u_n\}$, and subsets $S_1, \dots, S_m$ of U. W.l.o.g. we assume that each element of U is contained in at least one of the set $S_1, \dots, S_m$. We construct a bipartite graph G with vertex set $U \cup \{s_1, \dots, s_m\}$; we connect u_i with s_j exactly if $u_i \in S_j$, for each $1 \leq i \leq n$ and $1 \leq j \leq m$. We define the capacities by setting $c(s_j) = 0$ for each $1 \leq j \leq m$, and we set $c(u_i) = d(u_i) - 1$ for each $1 \leq i \leq n$.

First observe that for any solution R for (G, c) there exists a solution R' with $|R'| \leq |R|$ and $R' \subseteq \{s_1, \dots, s_m\}$. To see this, suppose that $u_i \in R$ for some i. It is easy to see that replacing u_i by any vertex $s_j \in N_G(u_i)$ still yields a solution. Hence, replacing every vertex in $U \cap R$ with corresponding vertices from $\{s_1, \dots, s_m\}$ we obtain a solution R' not larger than R, as desired.

Since each vertex of $G - U$ has zero capacity, we know that there must exist an orientation of G where all edges are oriented towards U except those incident to some vertex of R'. Such an orientation can respect the capacities for vertices of U only if each $u \in U$ has at least one outgoing arc, meaning that it is adjacent to at least one vertex in R'. By the definition of G, this implies that the set family $\{S_j \mid s_j \in R'\}$ yields a solution to the Set Cover instance I_{SC}. It is also easy to see that any solution for I_{SC} containing k sets yields a solution for (G, c) with k vertices.

Hence, an approximation for vdibo with ratio $\varepsilon \ln n$ where n is the number of vertices in the input graph would yield an approximation with ratio $\varepsilon \ln N$ for Set Cover on inputs of size N; using the result of Moshkovitz [23] the theorem follows. □

5 Conclusions

We conducted a thorough examination of the computational complexity of the MinStanding(S) problem both in the case where the set S of outcomes is general, obtaining mostly intractability results, and where S is complete. The latter case lead us to the investigation of a vertex-deletion problem concerning in-degree bounded orientations of graphs, for which in turn we discovered a strong connection to a special class of matroids (namely, gammoids). Based on this connection, we were able to provide a polynomial-time approximation for our problem with roughly optimal approximation ratio.

We hope our findings will motivate researchers to investigate further possibilities for revealing and utilizing connections between the area of classical graph (or matroid) theory and problems connected to sports competitions, or more generally, to social choice. In particular, it would be interesting to see which special cases of Matroid Base Cover may have practical applications, and whether these special cases admit efficient (approximation or optimal-value) algorithms.

References

1. Ilan Adler, Alan L. Erera, Dorit S. Hochbaum, and Eli V. Olinick. Baseball, optimization and the world wide web. *Interfaces*, 32(2):12–22, 2002.

2. Yuichi Asahiro, Jesper Jansson, Eiji Miyano, and Hirotaka Ono. Upper and lower degree bounded graph orientation with minimum penalty. In *CATS 2012: Proceedings of the 18th Comouting: The Australasian Theory Symposium*, volume 128 of *Conferences in Research and Practice in Information Technology*, pages 139–146, 2012.
3. Thorsten Bernholt, Alexander Gülich, Thomas Hofmeister, and Niels Schmitt. Football elimination is hard to decide under the 3-point-rule. In *MFCS 1999: Proceedings of the 24th International Symposium on Mathematical Foundations of Computer Science*, volume 1672 of *Lecture Notes in Computer Science*, pages 410–418. Springer, 1999.
4. Katarína Cechlárová, Eva Potpinková, and Ildikó Schlotter. Refining the complexity of the sports elimination problem. *Discrete Applied Mathematics*, 199:172–186, 2016.
5. M. Chrobak and D. Eppstein. Planar orientations with low out-degree and compaction of adjacency matrices. *Theoretical Computer Science*, 86:243–266, 1991.
6. Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, Heidelberg, 2005.
7. Yann Disser and Jannik Matuschke. Degree-constrained orientations of embedded graphs. *Journal of Combinatorial Optimization*, 31(2):758–773, 2016.
8. Rod G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, London, 2013.
9. Rod G. Downey and Micheal R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer-Verlag, New York, 1999.
10. Stefan Felsner. Lattice structures from planar graphs. *The Electronic Journal of Combinatorics*, 11:#R15, 2004.
11. András Frank and András Gyárfás. How to orient the edges of a graph. *Colloquia mathematica societatis János Bolyai*, 18:353–364, 1976.
12. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, 1979. A Series of Books in the Mathematical Sciences.
13. Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.
14. Dan Gusfield and Charles U. Martel. A fast algorithm for the generalized parametric minimum cut problem and applications. *Algorithmica*, 7(5&6):499–519, 1992.
15. Dan Gusfield and Charles U. Martel. The structure and complexity of sports elimination numbers. *Algorithmica*, 32(1):73–86, 2002.
16. S. L. Hakimi. On the degrees of the vertices of a directed graph. *Journal of the Franklin Institute*, 279:290–308, 1965.
17. Danny Hermelin, Matthias Mnich, and ErikJan Leeuwen. Parameterized complexity of induced H-matching on claw-free graphs. In *ESA 2012: Proceedings of the 16th Annual European Symposium on Algorithms*, volume 7501 of *Lecture Notes in Computer Science*, pages 624–635. Springer, 2012.
18. Alan J. Hoffman and Theodore J. Rivlin. When is a team "mathematically" eliminated? In *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 391–401. Princeton University Press, 1970.
19. Walter Kern and Daniël Paulusma. The new fifa rules are hard: complexity aspects of sports competitions. *Discrete Applied Mathematics*, 108(3):317–323, 2001.
20. Walter Kern and Daniël Paulusma. The computational complexity of the elimination problem in generalized sports competitions. *Discrete Optimization*, 1(2):205–214, 2004.

21. Eugene L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Dover Books on Mathematics. Courier Corporation, 2012.
22. S. Thomas McCormick. Fast algorithms for parametric scheduling come from extensions to parametric maximum flow. *Operations Research*, 47(5):744–756, 1999.
23. Dana Moshkovitz. The projection games conjecture and the NP-hardness of $\ln n$-approximating Set-Cover. *Theory of Computing*, 11:221–235, 2015.
24. Stefan Neumann and Andreas Wiese. This house proves that debating is harder than soccer. In *FUN 2016: Proceedings of the 8th International Conference on Fun with Algorithms*, pages 25:1–25:14, 2016.
25. J.G. Oxley. *Matroid Theory.* Oxford graduate texts in mathematics. Oxford University Press, 2006.
26. Hazel Perfect. Applications of Menger's graph theorem. *J. Math. Anal. Appl.*, 22:96–110, 1968.
27. Michael B. Richey and Abraham P. Punnen. Minimum perfect bipartite matchings and spanning trees under categorization. *Discrete Applied Mathematics*, 39:147–153, 1992.
28. Lawrence W. Robinson. Baseball playoff eliminations: An application of linear programming. *Operations Research Letters*, 10(2):67 – 74, 1991.
29. Benjamin L. Schwartz. Possible winners in partially completed tournaments. *SIAM Review*, 8:302–308, 1966.
30. Kevin D. Wayne. A new property and a faster algorithm for baseball elimination. *SIAM Journal on Discrete Mathematics*, 14(2):223–229, 2001.