Football Elimination is Hard to Decide Under the 3–Point–Rule
(Extended Abstract)

Thorsten Bernholt, Alexander Gülich, Thomas Hofmeister, Niels Schmitt

Lehrstuhl Informatik 2, Universität Dortmund, D–44221 Dortmund, Germany.
{bernholt,guelich,hofmeist,schmitt}@Ls2.cs.uni-dortmund.de

Abstract *The "baseball elimination problem" is a classic problem which has already been considered from the computational point of view in the 1960's. At some stage of the baseball season, there is a set of games which have already been played and there is another set of remaining games. The problem consists in determining for a given team whether or not they are already "eliminated", i.e., whether they can no longer become champions. Early solutions proposed a network flow approach which resulted in polynomial time algorithms. The interest in this kind of elimination problem was recently revived by Wayne [4] who proved an interesting threshold property which allows one to compute all eliminated teams simultaneously. Namely, there is a constant W^* such that a team is eliminated if and only if it can no longer obtain W^* or more points. Wayne also describes an algorithm for computing the threshold W^* in polynomial time. Gusfield and Martel [2] have generalized the proof of the existence of a threshold to a more general setting which includes European football, where the "3–point–rule" is in effect, i.e., 3 points are awarded for a win and 1 point is awarded for a tie.*

In this paper, we show that determining the elimination of a European football team under the 3–point–rule is $\mathcal{NP}$–complete. As a consequence, the generalized threshold result of Gusfield and Martel is of no use for the European football system since computing the corresponding threshold value is hard if $\mathcal{P} \neq \mathcal{NP}$. We also show that the elimination problem is still $\mathcal{NP}$–complete if all teams have at most three remaining games each while the problem can be solved in polynomial time if each team has at most two remaining games.

1 Introduction

A football/basketball/handball season consists of a set $\{1, \ldots, N\}$ of teams which have to play against each other. For every game between teams i and j, points are awarded according to some rule. Let us denote the "(α, β)–rule" to mean the following: If a game is won by one team, the winner gets α points and the losing team gets 0 points. If a game is a tie, then both teams get β points each. European football leagues are currently played under the $(3, 1)$–rule, which we also refer to as the "3–point–rule".

In (American) baseball, ties are not possible, i.e., every game has a winner which is awarded one point. We will refer to this system as the "1–point–rule".

At some stage of the season, each team has played a number of games and there is a number of remaining games. A team i is called "eliminated" if for all possible outcomes of the remaining games, there is at least one team which has more points than team i. Thus, eliminated teams are the ones which can no longer become champions. Correspondingly, we say that a team i "can still become champions" if there are outcomes for the remaining games such that no team has more points than team i.

The "baseball elimination problem" (which was already studied to some extent in the 1960's) is the problem of determining whether a given team i is eliminated under the 1–point–rule. Correspondingly, we can ask for the 3–point–rule whether a given team is already eliminated. We will refer to this problem as the "European Football Elimination Problem" (or EFEP for short).[⋆]

The motivation for considering these two problems is of course that they are coming from a "real–life application" and that they are also interesting from a combinatorial point of view.

The baseball elimination problem can be solved in polynomial time by network flow algorithms (see e.g. [1] and [4] for a more complete survey on the literature). Variations on the baseball elimination problem have also been considered, e.g., it was recently shown by McCormick [3] that it is $\mathcal{NP}$–complete, given t, to determine whether a given team can still achieve a position among the top t teams.

On the other hand, it is surprising that no corresponding results for the European Football Elimination Problem have been found so far. The results in this paper close this gap. The main result is that the European Football Elimination Problem under the 3–point–rule is an $\mathcal{NP}$–complete problem. It also follows from our proof that the restricted version of EFEP, where every team has at most three remaining games, is $\mathcal{NP}$–complete as well. On the other hand, we show how EFEP can be solved in polynomial time if every team has at most two remaining games. We only remark that our results can be generalized to other (α, β)–rules.

The interest in sports elimination problems was recently revived by Wayne [4] who investigated the complexity of determining the set of all eliminated teams simultaneously (under the 1–point–rule). The naive method computes this set by determining for every team whether or not they are eliminated. Wayne showed that there is a surprising property which facilitates the computation of the set of all eliminated teams. Namely, he showed that given a list of played and remaining games, there is a constant W^* such that the following holds: For all teams i (which have w_i points and g_i remaining games), it holds that team i is eliminated if and only if $w_i + g_i < W^*$. Wayne also showed how this threshold value W^* can be determined by a single preflow–push maximum flow computation on a network. Gusfield and Martel [2] have extended the existence result to the 3–point–rule, i.e., they have shown that there exists a corresponding constant W^*. It follows from our results that knowing about the existence of such a threshold value W^* is useless for EFEP, as W^* is hard to compute, unless $\mathcal{P} = \mathcal{NP}$.

Finally, let us notice the following: a few years ago, the European football leagues were played under the $(2, 1)$–rule. The solution to EFEP under the $(2, 1)$–

[⋆] We define EFEP in such a way that its output is "yes" if a team can still become champions and "no" otherwise.

rule was simple: Since the number of points awarded to teams in one game was a constant (namely, 2), independent of the outcome of the game, it could be regarded as a special instance of the baseball elimination problem. Namely, one game between teams i and j under the $(2, 1)$–rule could be seen as two games between teams i and j under the 1–point–rule (without ties). Hence, EFEP under the $(2, 1)$–rule can be solved by network flow algorithms and it is also amenable to the new result of Wayne. Our paper has the consequence that for the newly introduced $(3, 1)$–rule, it is much harder to decide whether a team can still become champions.

2 Preliminaries

Assume that an instance of the European Football Elimination Problem consists of N teams. The input contains a list $p_1, \ldots, p_N$, where p_i is an integer representing the points that team i have been awarded in the games they have played so far. The input also contains a list of games that still have to be played. W.l.o.g., we can assume that the European Football Elimination Problem asks whether team number N can still become champions. Furthermore, we assume that team N has no remaining game to play. The reason why we can do so is that we can assume w.l.o.g. that team N wins all of its remaining games, should there be any.

We now represent an input instance to EFEP by an undirected multigraph which contains N labeled vertices $1, \ldots, N$. Vertex i stands for team i and it is labeled with the number $p_i - p_N$. By this choice, a negative label on vertex i means that team i has less points than team N and a positive label means that team i has more points than team N. The edges are constructed as follows: An edge between vertex i and vertex j stands for one game between the teams i and j that still has to be played. If there is more than one remaining game between teams i and j, there is a corresponding number of edges between vertices i and j. Games that are already played have no corresponding edge in the graph. (Thus, vertex N is an isolated vertex which has label 0.)

In order to make our proofs technically easier, we will make the following slight modification which we refer to as "modification $(*)$". We modify the rule of how to award points as follows: For a game between teams i and j that is not yet played, both teams get one point. This modification does not have any influence on the problem since at the end of a season, when all games are played, the points awarded to the teams are not changed under this new rule. We will assume that modification $(*)$ is already taken into account in the input to the European Football Elimination Problem.

3 The Reduction

In this section, we prove the following result:

Theorem 1. *The European Football Elimination Problem is $\mathcal{NP}$–complete under the $(3, 1)$–rule.*

It is clear that EFEP is in $\mathcal{NP}$, since we can guess the outcomes of the remaining games and compute the final ranking. For showing the $\mathcal{NP}$–completeness, we reduce the well–known $\mathcal{NP}$–complete problem 3–SAT to EFEP. The reduction is easily seen to be polynomial–time.

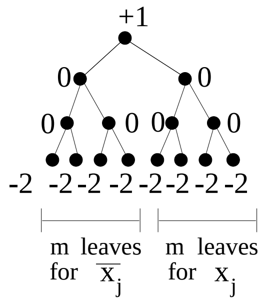
We proceed as follows: for each input formula to 3–SAT, we construct a labeled multigraph which corresponds to an input to EFEP. First, we show that the 3–SAT–formula is satisfiable if and only if team number N can still become champions. We then show that the situation described by the labeled multigraph can in fact arise during a football season.

An input to 3–SAT is given by a formula in conjunctive normal form where each clause contains exactly three literals on different variables. We assume that the formula contains m clauses on the variables $x_1, \ldots, x_n$. By possibly copying clauses, we can assume that $m \geq 2$ is a power of two. The problem 3–SAT asks whether the input formula has a satisfying assignment from $\{0,1\}^n$ to the variables $x_1, \ldots, x_n$. It is known to be an $\mathcal{NP}$–complete problem.

As an example, the formula $(x_1 \vee \overline{x_3} \vee \overline{x_4}) \wedge (x_3 \vee x_2 \vee \overline{x_1})$ might be an input to 3–SAT which has a satisfying assignment $(1, 1, 0, 0)$.

Given an input formula for 3–SAT, we construct an input to the European Football Elimination Problem by describing the labeled multigraph. In the construction of the multigraph, we use the following components:

For every variable x_j, we have a full binary tree (called "x_j–tree") which looks as follows. (The number m of clauses determines the depth of the tree.)



The x_j–tree has exactly $2m$ leaves. The leaves in the left subtree of the root are called "$\overline{x_j}$–leaves", while the others are referred to as the "x_j–leaves". The root is labeled with $+1$, the leaves are labeled with -2, the other (inner) vertices are labeled with 0.

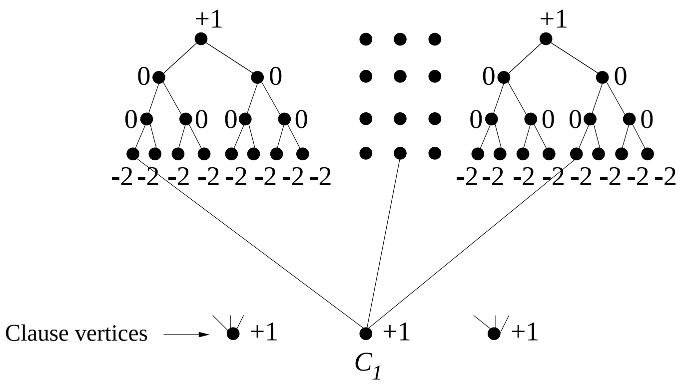
Let us appeal to the intuition why this tree is of use in our reduction. Call the vertex at the top of that tree A and its two children B and C.

If team N are to become champions, then we have to achieve that no team has more points than team N, i.e., no vertex should have a label larger than 0 at the end of the season. In particular, we have to make sure that vertex A no longer has a label larger than 0. How can we achieve this? Team A has two remaining games, namely, against its two children B and C. According to modification $(*)$, team A did already get one point for each of those games. Thus, we have to make sure that team A loses at least one of its two remaining games. Assume that A loses its game against team B. This means that the label of team B is increased to 2, since team B now gets three points for the game against A while previously, they only got one point for this game.

If team N are to become champions, we have to make sure now that the label of team B is not larger than 0 at the end of the season. With arguments similar to the ones we just gave, it follows that team B has to lose both its games against its two children. The argument can be repeated, i.e., we have an avalanche effect, where all the inner vertices have to lose their games against their children. This avalanche can obviously only be stopped in the leaves of the tree, when the labels of the leaves are increased from -2 to 0.

For the intuition, we remark that the decision whether team A in the x_j–tree loses against team B or against team C corresponds to the decision whether to assign $x_j = 0$ or $x_j = 1$.

Let us now continue to describe the reduction. For every clause number i, we introduce an extra vertex C_i which is labeled with $+1$ and which we connect with three vertices. Namely, if this i–th clause contains the literal x_j, then we connect C_i with the i–th $\overline{x_j}$–leaf. If the i–th clause contains the literal $\overline{x_j}$, then we connect C_i with the i–th x_j–leaf. Finally, we add an isolated vertex with label 0, for "team N". The reduction can be visualized as follows:



We first show that for the above input instance to EFEP, the following holds: Team N can become champions if and only if the 3–SAT–formula is satisfiable. We then have to show that the input instance can in fact arise during a season, i.e., we have to show that the configuration of points and remaining games can in fact arise by appropriate outcomes for the games that have already been played.

Lemma 1. *Team N can become champions in the constructed EFEP–instance if and only if the input 3–SAT–formula is satisfiable.*

Proof. Assume that the input 3–SAT–formula is satisfiable and let $(a_1, \ldots, a_n) \in \{0,1\}^n$ be a satisfying assignment. We choose the following outcomes for the remaining games:

If a_j=0, then we declare all edges (=games) in the left subtree of the x_j–tree to be won by the child. We declare all edges in the right subtree to be ties. This has the following effect on the labels: The labels in the right subtree do not change, and the label of the root and all vertices in the left subtree is 0.

If a_j=1, then we proceed in the same way, with the roles of the left and right subtrees interchanged.

The only remaining games are between the clause vertices C_i and some leaves.

Since the assignment $(a_1, \ldots, a_n)$ is a satisfying assignment, we know that for every clause, there is one literal in the clause which is assigned a 1. Consider clause number i and assume that it was satisfied by the assignment $x_j := 1$. We declare the outcomes of the games of vertex C_i as follows: C_i loses its game against the $\overline{x_j}$–leaf and its other two remaining games are ties.

The label of C_i changes to 0, and since x_j was assigned a 1, the label of the i–th $\overline{x_j}$–leaf was -2 and is now 0, while the other two leaves that C_i is incident to, have their labels unchanged. Altogether, we obtain that a satisfying assignment to the 3–SAT–formula can be turned into outcomes for the remaining games such that all teams/vertices have a label of 0 or smaller, i.e., team N can still become champions.

We now show the other direction, i.e., given outcomes of the remaining games such that no team has more points than team N, we construct a satisfying assignment for the 3–SAT–formula.

We first examine the root of an x_j–tree. By the remarks before Lemma 1, the root loses at least one of its two remaining games. Assume that the root loses against the left child. Due to the avalanche effect, all inner vertices of the left subtree lose their games against their children.

We now define an assignment as follows: If the root of the x_j–tree loses against its left child, then let $x_j = 0$. Otherwise, let $x_j = 1$. This yields a satisfying assignment for the 3–SAT–formula, as we show now.

If it was not a satisfying assignment, then there would be at least one unsatisfied clause. Assume that clause i is not satisfied. If clause i contains a literal x_k, then the corresponding edge is connected with a $\overline{x_k}$–leaf. Since clause i is not satisfied, variable x_k was set to 0 and thus the label of the $\overline{x_k}$–leaf must be zero (due to the avalanche effect). Thus, vertex C_i has three outgoing edges which all enter leaves which have label 0.

Thus, it is not possible that C_i loses any of its three remaining games, because otherwise one of those leaves would have a label which is larger than 0. This means that vertex C_i has a label of at least $+1$, and team N cannot become champions which is a contradiction. □

Constructing the set of games already played

We now show that the above configuration can in fact occur during a season. I.e., we find a list of already played games and outcomes for them such that the situation of the season is represented by the multigraph described in the reduction. Let N be the number of vertices resulting from the described reduction. We will embed the problem into a season with a larger number of teams, where the additional dummy teams just serve the purpose of allowing us to adjust the labels of the teams we are interested in.

Let $V = \{v_1, \ldots, v_N\}$ be the set of the N teams which occurred in the reduction previously described. We introduce two new sets of teams $V' = \{v'_1, \ldots, v'_N\}$ and $V'' = \{v''_1, \ldots, v''_{2N}\}$.

At the beginning of a season, we assume that every team from the set $V \cup V' \cup V''$ has to play against each other team from the set $V \cup V' \cup V''$ exactly $t \geq 1$ times. If $t > 1$, then we can, for each pair of teams i and j, declare $t-1$ games between teams i and j to be a tie. Then all teams have an equal number of points, and for every pair of teams i and j, there is exactly one game remaining between the two. Thus, in the following, we can restrict ourselves to seasons with t=1.

We now want to describe a situation during the season which leads to the multigraph described in the reduction. For this purpose, we declare which games have already been played by describing the outcomes of the games as follows: The outcome of a game between team i and team j is:

$$\begin{array}{lll} \text{a win for team } i & \text{if } i \in V' & \text{and } j \in V, \\ \text{a win for team } i & \text{if } i \in V & \text{and } j \in V'', \\ \text{a tie} & \text{if } i \in V' & \text{and } j \in V'', \\ \text{a tie} & \text{if } i \in V' & \text{and } j \in V', \\ \text{a tie} & \text{if } i \in V'' & \text{and } j \in V''. \end{array}$$

The only remaining games are the games between teams in the set V. Due to modification $(*)$, it does not make a difference if we declare some of the games in the set V as games that were already played and that ended in a tie. Thus, we can achieve that the remaining games are exactly the remaining games between teams from V which are needed in the reduction.

According to the above described outcomes, we have the following situation:

$$\begin{array}{lll} \text{Teams from } V \text{ have} & (N-1) + 3 \cdot (2N) & = 7N-1 \text{ points.} \\ \text{Teams from } V' \text{ have} & 3 \cdot N + (N-1) + 2N & = 6N-1 \text{ points.} \\ \text{Teams from } V'' \text{ have} & (2N-1) + N & = 3N-1 \text{ points.} \end{array}$$

For example, the $7N-1$ points for a team from V result from the observation that the $N-1$ games against the other teams from V are not yet played (or a tie), which, due to modification $(*)$, makes up for $N-1$ points. On the other hand, there are $2N$ wins against the teams from V''. The other points can be verified similarly.

The team for which we want to decide whether it is eliminated or not is in the set V. Hence, the labeled multigraph corresponding to the above situation of the season has labels 0 on the vertices from V and labels $-N$ and $-4N$ on the vertices of V' and V'', respectively.

The reduction described earlier requires that some of the teams in V have labels $+1$ and -2 instead of 0. We achieve this by using the dummy teams from V' and V'' as follows:

If a team $v_i \in V$ is to have a label $+1$, then we declare the game between v_i and v'_i to be a tie instead of a win for v'_i. This increases the label of v_i from 0 to $+1$ and reduces the label of v'_i by 2.

If a team $v_i \in V$ is to have a label -2, then we declare the game between v_i and v_i'' to be a tie instead of a win for v_i. This reduces the label of v_i from 0 to -2 and increases the label of v_i'' by 1.

After these changes, the labels of all teams $v \in V$ are as required in the reduction.

We have shown that during a season, the situation needed described in the reduction can in fact occur as a subproblem in a larger season. We only have to argue that the two problems are equivalent.

Note that in the above season with $4N$ teams, the team for which we want to decide elimination has $7N-1$ points. Hence the champions also will have at least $7N-1$ points. The teams from V' and V'' have at most $\max\{6N-1, 3N\} < 7N-1$ points and the only remaining games are between teams from V. Thus, EFEP on the set $V \cup V' \cup V''$ is identical to EFEP on the set V.

4 Restricted Versions of EFEP

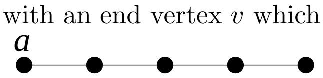
Taking a close look at the reduction in Section 3, we see that there is a degree bound: Every team has at most three remaining games to play. Thus, our reduction also shows that the restricted European Football Elimination Problem, where all teams have at most three remaining games, is already $\mathcal{NP}$–complete.

Naturally, the question arises what the situation is like if there are at most two remaining games for each team. We obtain the following result:

Theorem 2. *The European Football Elimination Problem under the* $(3, 1)$*–rule can be solved in polynomial time if there are at most two remaining games for each team.*

Proof. Consider the labeled input multigraph which describes the given instance of EFEP. Since every vertex has degree bounded by two, the graph is the union of components which are paths and cycles (and isolated vertices). It is enough to decide (independently of each other) for each of those paths and cycles whether there is an outcome for the remaining games such that all teams have a label which is not larger than zero.

First, consider a path with an end vertex v which has label a.



If $a \geq 2$, then team N can no longer become champions, since the label of v will be at least $+1$.

If $a = 1$, then it is clear that in order to make team N champions, the team corresponding to v has to lose its only remaining game. Hence, we can reduce the path by removing v and increasing the label of its only neighbor by two. Similarly, the following operations can be applied:

If $a \in \{0, -1\}$, then remove v and define the incident edge (=game) to be a tie. Team N can become champions if and only if it can become champions for the reduced path. (A moment's thought shows that the tie is the best result possible for team N.)

If $a \leq -2$, then remove v and define the incident edge (=game) to be a win for the removed team. Team N can become champions if and only if it can become champions for the reduced path.

Thus, in a linear number of steps, we can reduce a path to a trivial graph for which EFEP can be decided easily.

Now, consider a component which is a cycle. We choose one of the edges of the cycle and consider all three possible outcomes for the corresponding game. For each of the possible outcomes, we remove the edge from the cycle and adjust the labels correspondingly. We obtain a path which we can deal with as described above. Altogether, we can decide for every component in linear time whether there is an outcome for the remaining games such that all teams in the component have a label at most 0. $\square$

5 General (α, β)–rules

The $\mathcal{NP}$–completeness result which we have presented above can also be generalized to the case of the (α, β)–rule, where $\alpha > 2\beta$ (and $\beta \geq 1$). This can basically be achieved by replacing a label $+1$ in the reduction by a label β and by replacing a label -2 by $(\beta - \alpha)$, while labels 0 remain the same.

We are also able to prove the $\mathcal{NP}$–completeness for the (α, β)–rule, when $\beta < \alpha < 2\beta$ (and $\beta \geq 1$). Here, we have to use a different multigraph in the reduction, but the technique still is very similar to the one that we have presented. We omit the details of the proof.

References

1. R. K. Ahuja, T. L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.
2. D. Gusfield and C. Martel, *The Structure and Complexity of Sports Elimination Numbers,* Tech. Report CSE–99–1, CS Department, Univ. of California, Davis, January 1999.
3. S. T. McCormick, *Fast algorithms for parametric scheduling come from extensions to parametric maximum flow.* In: Proc. of the 28th Annual ACM Symp. on the Theory of Computing (STOC), 319–328, 1996.
4. K. D. Wayne, *A New Property and a Faster Algorithm for Baseball Elimination,* In: Proc. of the 10th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA), 1999. Preprint available as a link under the URL `http://www.cs.princeton.edu/~wayne/research.html`