Complexity of Scheduling and Predicting Round-Robin Tournaments

Dorothea Baumeister
Heinrich-Heine-Universität Düsseldorf
Düsseldorf, Germany
d.baumeister@uni-duesseldorf.de

Tobias Hogrebe
Heinrich-Heine-Universität Düsseldorf
Düsseldorf, Germany
tobias.hogrebe@uni-duesseldorf.de

ABSTRACT

Tournaments are commonly used to identify winners, for example in sports. We study the computational complexity of problems related to scheduling the matches in a tournament and predicting the outcome with a special focus on round-robin tournaments, which is the most prominent tournament type used in sports. Besides the general financial and intrinsically motivated interest of various agents in tournament prediction, the recently very relevant winner determination for suddenly discontinued tournaments is a strong motivation. We show the immense theoretical complexity of predicting the winners in round-robin tournaments even under the assumption that only three matchdays remain to be played. On the other hand, we present an FPT algorithm and analyze its practical complexity using experiments on real-world and generated data, showing the applicability of the algorithm in praxis. To the best of our knowledge, this is the first exact and not purely brute-force oriented approach for predicting round-robin tournaments.

KEYWORDS

Computational Complexity; Round-Robin Tournaments; Predicting; Scheduling; Probabilistic Tournaments

ACM Reference Format:
Dorothea Baumeister and Tobias Hogrebe. 2021. Complexity of Scheduling and Predicting Round-Robin Tournaments. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

Throughout almost the entire written history of humanity, sport has always played a special role. From children playing in the schoolyard all the way to amateur and professional sports, this special form of entertainment and recreation continues to attract immense interest around the world. In this work, we examine the computational complexity of the so-called evaluation problem, which asks how likely it is for a certain team to win a given tournament in which teams score points according to the outcomes of a match. We assume that some of the matches may already have been played, i.e., that we are in the course of a season, and that we are given a probability distribution over the outcomes of the remaining matches. Here we focus especially on round-robin tournaments, the most prominent form of sports tournaments followed by knockout tournaments. The study of this problem predates the emergence of modern complexity theory. In 1929, Zermelo [31] already studied the problem regarding round-robin tournaments in chess. In a round-robin tournament each team plays a fixed number of matches against each other team. Round-robin tournaments are used for many different championships in various sports, such as baseball, bridge, chess, hockey, and football. In addition, round-robin tournaments are also used as an intermediate stage in other types of tournaments such as the group stage of the FIFA World Cup and the UEFA Champions League. One reason for the frequent use of this tournament format is fairness, since it does not rely as heavily on the initial seeding as knockout tournaments, and that the final table tends to reflect the true ability levels of the participants (see, e.g., Ryvkin and Ortmann [26]). However, round-robin tournaments also have certain drawbacks, such as a high number of matches, as well as the fact that the champion and relegated teams could be determined several matchdays before the end of the tournament, which reduces the attractiveness. The latter brings us back to our original motivation for the evaluation problem. As we will discuss later, there is a large number of agents who have various financial, competitive, and/or intrinsically motivated reasons to be interested in predicting the outcome of a tournament. Beside sports, round-robin tournaments are also used in other areas. The most prominent example for this is the Copeland voting rule used in decision making, applicable in many types of applications, with possibly a high number of alternatives, such as planning (Ephrati and Rosenschein [13]), rank aggregation in web search (Dwork et al. [12]), recommender systems (Ghosh et al. [16]), and email classification (Cohen et al. [7]). For these applications, the problem can be used, for example, to include uncertainties about pairwise comparisons due to perturbations or to allow the evaluation of incomplete data. The evaluation problem in decision making was most prominently examined by Hazon et al. [18] and subsequently by Baumeister and Hogrebe [2], among others.

To the best of our knowledge, the computational complexity of the evaluation problem with respect to different types of tournaments was first and almost solely studied by Mattei et al. [22], with Saarinen et al. [27] later adding the hardness result for round-robin tournaments. We will discuss these results later in comparison to our results. For the elimination problem, the special case of the evaluation problem where we only ask if a team can win, there are a lot of exact approaches which offer an efficient solution in practice, e.g., ILP formulations (Robinson [24]), FPT algorithms (Cechlárová et al. [5]), and in some cases even polynomial-time algorithms (Kern and Paulusma [20], Schwartz [29]). However, to the best of our knowledge, for the evaluation problem itself there is so far only the possibility of an efficient but approximate solution using Monte Carlo simulations or the exact calculation using a

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

brute-force approach, i.e., the consideration of every possible combination of outcomes, which is far too time-consuming, even for very restricted real-world instances. A major problem regarding Monte Carlo simulations is that, by their very nature, they can easily overlook scenarios with low probability. This means that for teams with a very low probability of ending up as the champion, the relative error can easily become infinite.

Therefore, our main contribution is the development of an exact algorithm with an execution time suitable for practical application. Furthermore, we provide theoretical results for the completion of partial schedules, show that evaluation is computationally difficult even assuming strong constraints on the number of remaining matchdays, and analyze the practical relevance of our algorithm by experiments on real-world and generated data.

2 PRELIMINARIES

In the following, we present the basic definitions for the types of tournaments considered here. A *tournament* $\mathcal{T} = (T, M)$ consists of a set of teams $T = \{t_1, \dots, t_n\}$ with $n \geq 2$ and a set of matches $M = \{m_1, \dots, m_g\}$ with $g \geq 1$ over T. If not further specified, n corresponds to the given number of teams. Each match m_h, $1 \leq h \leq g$, is associated with a tuple of competing teams $(t_i, t_j) \in T \times T$ with $t_i \neq t_j$ and will be denoted by $m_h : (t_i, t_j)$. Further, a set of possible *outcomes* $O = \{(\alpha_1, \beta_1), \dots, (\alpha_\ell, \beta_\ell)\}$ with $\ell \geq 1$ and $\alpha_s, \beta_s \in \mathbb{N}_0$ for $1 \leq s \leq \ell$ is given, with at least one $(\alpha_s, \beta_s) \in O$ with $\alpha_s \neq \beta_s$. We call a set of possible outcomes *symmetric*, if it holds that $(\beta_s, \alpha_s) \in O$ for all $(\alpha_s, \beta_s) \in O$. To determine the winners of a tournament, each match is assigned to an outcome. If a match $m : (t_i, t_j)$ is assigned the outcome (α_s, β_s), this means that team t_i receives α_s points and team t_j receives β_s points. The *winners* are the teams with the maximum number of points. We refer to the undirected multigraph consisting of the teams in T as nodes and the edges representing the matches in M as the *match graph*. A *round-robin tournament* with k rounds is a concatenation of k single-round round-robin tournaments in which each team plays exactly once against every other team. In accordance with the majority of real-world instances, we assume that the number of teams n is even in the context of round-robin tournaments.

There are many examples for round-robin tournaments in practice. In football (soccer), usually $k = 2$ rounds are played using the *3-point rule* with $O = \{(3, 0), (1, 1), (0, 3)\}$. Other examples include baseball with $O = \{(1, 0), (0, 1)\}$, basketball with $O = \{(2, 0), (1, 1), (0, 2)\}$, chess with $O = \{(1, 0), (1/2, 1/2), (0, 1)\}$, and volleyball with $O = \{(3, 0), (2, 1), (1, 2), (0, 3)\}$, with the number of rounds k usually depending on the number of teams. The Copeland$^\alpha$ voting rule with $\alpha \in \mathbb{Q}$, $0 \leq \alpha \leq 1$ is equivalent to a round-robin tournament with $k = 1$ and $O = \{(1, 0), (\alpha, \alpha), (0, 1)\}$. While some of the sets of possible outcomes presented above include rational numbers, all of them can be scaled up to equivalent sets over $\mathbb{N}_0$.

Computational Complexity. We assume that the reader is familiar with the basic concepts of computational complexity, such as P and NP, their counting/function problem counterparts FP and #P, fixed-parameter tractability (FPT), and polynomial-time many-one and Turing reducibility. For more information, we refer to the textbooks by Arora and Barak [1] and Papadimitriou [23].

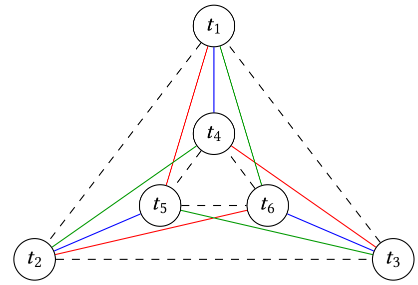


Figure 1: Example of a premature partial schedule for a round-robin tournament with six teams.

2.1 Scheduling

In a real-world tournament, the matches are usually organized into clearly separated matchdays with each team playing at most once on every matchday. We refer to such a partition of a tournament as a *schedule*, which is formally given by an edge coloring $S : M \rightarrow D$ of the match graph with $S(m) \neq S(m')$ for any adjacent matches $m, m' \in M$, where $D = \{1, \dots, d\}$ denotes the set of available matchdays. For a tournament $\mathcal{T}$ we call a schedule S *optimal* if it uses only the minimum number of matchdays required.

Constructing an optimal match schedule can be hard for general tournaments, as the problem of checking if a graph with maximum degree Δ is edge-colorable with Δ colors, which is the minimum number necessary, or only with $\Delta + 1$ colors, which is the maximum number necessary, is already NP-complete as shown by Holyer [19]. The problem of creating an optimal match schedule for a round of a round-robin tournament, as considered here, can be solved efficiently, since the complete graph can be colored with $\Delta = n - 1$ colors for even n (see Behzad et al. [3]). However, we will encounter the problem of embedding a *partial schedule*, i.e., a schedule covering only a certain number of matchdays, in a complete schedule for a round-robin tournament. Rosa and Wallis [25] studied this problem and refer to the property of a partial schedule not being extendable to a complete schedule as *premature*. We refer to the respective decision problem of whether a partial schedule can be extended to an optimal schedule as NON-PREMATURE.

Motivation. One motivation to study the problem of premature partial schedules is the observation by Rosa and Wallis [25] that scheduling a round-robin tournament by consecutively planning the matchdays by random or even intentional can lead to a non-optimal number of matchdays. Figure 1 shows for example a partial schedule with $r = 3$ prescheduled matchdays for $n = 6$ teams which cannot be completed using $\Delta = 5$ colors. The problem is particularly interesting for the organizing associations which have to create the schedules based on various criteria such as fairness, attractiveness, and feasibility with respect to various restrictions.

2.2 Evaluation

In the following, we will define the evaluation problem for tournaments. Assume we are given a set of possible outcomes O. Given a tournament $\mathcal{T} = (T, M)$, we refer to a collection of probability functions $\rho = (\rho_m)_{m \in M}$ over O, one probability function for each match in M, as an *outcome probability profile*. We assume by default that those functions are given as simple lists, associating each outcome with the respective probability given as a rational number. The evaluation problem is defined as follows.

O-Evaluation	
Given:	A tournament $\mathcal{T} = (T, M)$, an outcome probability profile ρ regarding $\mathcal{T}$ over O, and a team $p \in T$.
Question:	What is the probability that p ends up as the unique winner of the tournament?

We refer to the evaluation problem restricted to round-robin tournaments with a given optimal schedule as part of the input as O-RRTS-Evaluation. While we consider the evaluation problem here with respect to being the unique winner, and therefore regarding the championship, it can of course be extended to other events such as qualification and relegation. For a given instance of O-Evaluation, a match $m \in M$ is referred to as *open* if the outcome probability function ρ_m assigns a positive probability to two or more outcomes. Assuming that the tournament is a round-robin tournament and given an additional schedule, we call a matchday open if it contains at least one open match. Referring back to our real-world setting, if we assume that we are at a certain point in the course of a tournament, none of the matches that have already been played can still be open, since their outcomes have already been determined. On the other hand, we allow matches with a certain outcome to take place on open matchdays. The reductions presented here can easily be adjusted to ensure that all matches on the open matchdays have ambiguous outcomes by choosing a probability, depending on the instance, very close to one instead of one and distributing the remaining probability to another outcome.

Motivation. It is straightforward to see that the tournament evaluation problem is of great interest to many different types of agents in various sports. For the media and fans, there is a general intrinsically motivated interest in predicting who might become the champion, for fans especially when it comes to their own team. For the teams, their managements, and sponsors, there is both a competitive and financial interest. Predictions about the outcome of a season can be used to develop crossmatch strategies such as sparing certain players, making personnel decisions such as transfers, and planning travels and investments. Another huge area of agents with a particularly high financial interest in the outcomes of seasons is sports gambling, including both the bidders themselves as well as the bookmakers. The evaluation problem also provides the computational framework for calculating the so-called *match importance* as introduced by Scarf and Shi [28], which measures the importance of a single match for the outcome of a season. Again, the motivations vary from the allocation of referees to the decision of broadcasters which matches they will show. Related to importance is the study of *match relevance* (see Faella and Sauro [14]), which is focused on the design of tournaments in which as many matches as possible still have relevance for the final standing in the course of the tournament. The aforementioned motivations are reinforced by the fact that a large amount of data is publicly available. Schedules for the current season are usually published by the associations before the start of the season, as well as the current results. Schedules and results from previous seasons are publicly offered on various pages. An easy to aggregate source for probability profiles are betting odds, which have a good predictive quality, as demonstrated by Spann and Skiera [30]. Another more complex but not necessarily better approach is the use of statistical models based on historical data and current factors.

An additional motivation for the evaluation problem, which has recently become very relevant, is the discontinuation of seasons due to certain circumstances. In practice, the championship and qualifications were often decided by using the scores normalized according to the number of matches. However, this method can be highly problematic and unfair. Imagine, for example, that the team currently in the lead by a narrow margin would only have had matches against strong teams, while the team in second place would only have had matches against weak teams. In these cases, the evaluation problem can be used to justify or criticize decisions.

3 RESULTS

First, we consider the scheduling of round-robin tournaments.

3.1 Scheduling

Colbourn [8] showed that Non-Premature is NP-complete. However, this hardness does not extend to the case where the problem is parameterized by the number of already scheduled matchdays.

Theorem 3.1. Non-Premature *is* FPT *when parameterized by the number of prescheduled matchdays.*

The theorem can be shown using the result of Chetwynd and Hilton [6, Corollary 1], that a regular graph with an even number of nodes n and degree $\Delta \geq 6n/7$ can be edge colored with Δ colors. It follows that a partial schedule with r prescheduled matchdays cannot be premature for $n > 7(r + 1)$ teams. Thus, parameter r limits the number of teams up to where one can use a brute-force approach and where the answer to the problem becomes trivial.

While this result is positive in that we can embed any combination of r prescheduled matchdays into a complete optimal schedule for sufficiently large n compared to r, this only holds if the prescheduled matchdays are complete. For example, consider the schedule S with $S(m) = 1$ for $m : (t_{2j+1}, t_{2j+2})$ for $j \in \{1, \ldots, n/2 - 1\}$ and $S(m) = 2$ for $m : (t_1, t_2)$ over a round-robin tournament $\mathcal{T} = (T, M)$ with $T = \{t_1, \ldots, t_n\}$ and even n. We refer to a partial schedule with possibly incomplete matchdays as *sparse*. Sparse schedules are of particular interest if the organizers are interested in spreading certain matches, like top matches, over several matchdays. In the following, we show that sparse schedules with a certain number of fixings per matchday can always be extended to partial schedules.

Theorem 3.2. *Every sparse schedule over r matchdays with at most $n/2 - r$ matches each can be extended to a partial schedule*

Proof. Assume we are given a round-robin tournament $\mathcal{T} = (T, M)$ with n teams and a sparse schedule over r matchdays. Without loss of generality, we assume that the r partially prescheduled matchdays are $\{1, \ldots, r\}$. We denote by $g_1, \ldots, g_r$ the number of prescheduled matches on the respective matchday with $g_j \leq n/2 - r$ for $1 \leq j \leq r$. Now, we provide a scheme to construct a partial schedule for r complete matchdays for the given sparse schedule.

We complete matchday $j = 1, \ldots, r$ one after the other by considering the match graph G_j of $\mathcal{T}$ restricted to teams without prescheduled matches on matchday j and to matches not appearing on any other prescheduled matchday with respect to the current state of the schedule. Thus, G_j contains $n_j = n - 2 \cdot g_j$ teams and the minimum degree is $\Delta_j = n_j - 1 - (r-1) = n_j - r$. As $g_j \leq n/2 - r$, it follows that $\Delta_j \geq n_j/2$. If $n_j = 0$, the matchday is already complete. If $n_j = 2$, we can simply add the remaining match to complete matchday j. If $n_j \geq 4$, we can apply Dirac's Theorem on Hamiltonian cycles (Dirac [11]) to find a Hamiltonian cycle in G_j as $\Delta_j \geq n_j/2$, which can be decomposed into a perfect matching. Thereby, we can complete matchday j by adding the matches contained in the matching. After completing each matchday, we receive a partial schedule extending the sparse schedule. □

The partial schedule created in this way, and thus also the original sparse schedule, cannot be premature if the number of prescheduled matchdays r is not too large compared to n. For example, for $n > 7(r+1)$ following from the previously mentioned result by Chetwynd and Hilton [6] or $n \geq 2r$ for sufficiently large n following from the results by Csaba et al. [9]. Bernholt et al. [4] showed that the elimination problem for the 3-point rule is NP-complete for round-robin tournaments in which each team has at most three open matches. Using Theorem 3.2, the previously mentioned results regarding prematurity, and additional teams, this result can be extended to the case where we also require the existence of an optimal schedule with at most three open matchdays.

3.2 Evaluation

In this section, we will elaborate the following central dichotomy result regarding the complexity of the evaluation problem: For a fixed symmetric set of outcomes O, it holds that O-RRTS-Evaluation is in FP for at most two open matchdays and #P-hard otherwise. We start by showing the hardness result.

Theorem 3.3. *Given a fixed symmetric set of outcomes O and a fixed integer $r \geq 3$. O-RRTS-Evaluation is #P-hard, even assuming that there are only r open matchdays and all outcome probabilities are from $\{0, 1/2, 1\}$.*

Previous results in the literature regarding the hardness of predicting round-robin tournaments (see Saarinen et al. [27]) do not take into account the existence of schedules and require a variable number of open matches for certain teams. This greatly limits the possible points of observation in the course of a tournament and makes it even theoretically impossible to encounter the given or a similar situation in the course of a real-world tournament.

Proof. Dagum and Luby [10] showed that #Perfect-Bipartite-Matching, the problem of counting perfect matchings in a bipartite graph, is #P-complete for r-regular bipartite graphs with fixed $r \geq 3$. Assume, we are given a fixed symmetric set of outcomes O and a fixed integer $r \geq 3$. We reduce the problem of counting perfect matchings in an r-regular bipartite graph to O-RRTS-Evaluation requiring r open matchdays using a polynomial-time Turing reduction, proving the #P-hardness of the respective evaluation problem.

Given an instance of #Perfect-Bipartite-Matching consisting of an r-regular bipartite graph $G = (U, V, E)$ with $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$, and edges $E \subseteq U \times V$. The reduction is as follows. From the symmetrical set of outcomes O we choose an arbitrary symmetrical pair of outcomes (α, β) and (β, α) with $\alpha \neq \beta$. Since these two outcomes will be the only outcomes that will receive a positive probability in the constructed instance, we will assume for the sake of simplicity and without loss of generality that $(\alpha, \beta) = (1, 0)$ holds. We say that a team *wins* a match if it receives the point. The constructed evaluation instance consists of a round-robin tournament $\mathcal{T} = (T, M)$ with the set of teams $T = U \cup V \cup C \cup D \cup \{h, p\}$ with $C = \{c_1, \ldots, c_r\}$, $D = \{d_1, \ldots, d_{r-2}\}$, and additionally $\{g_1, g_2\}$ if $n + r$ is odd, and M as the respective set of matches of a single-round round-robin tournament over T, and an outcome probability profile ρ regarding $\mathcal{T}$. Later, we will explain how to extend the proof to round-robin tournaments with an arbitrary number of rounds. For the sake of readability, we assume that $n + r$ is even. When $n + r$ is odd the reduction has to be adapted such that all teams win against g_1 and g_2 and the construction of the schedule requires a slight adjustment to include g_1 and g_2.

In the following, we will construct the outcome probability profile ρ. First, we will fix all matches in $M_{\bar{E}} \subset M$, the set of matches that do not correspond to edges of G. Here, *fixing a match* means that we assign an outcome to a match with probability 1. First, we consider all matches in $M_{\bar{E}}$ between two teams in $U \cup V$ and additionally the matches between a team in $U \cup V$ and a team in $C \cup \{p\}$. In this set of matches, each team appears exactly $2n$ times. Therefore, we can use an arbitrary Eulerian orientation of the match graph regarding those matches to fix the matches in a way that each team wins exactly n matches. There are several polynomial-time algorithms to construct such an orientation of a graph, for an overview see Fleischner [15]. Second, the matches in $M_{\bar{E}}$ between two teams in $C \cup D$ are fixed in a similar fashion. We use an arbitrary Eulerian orientation of the match graph regarding the matches between the teams in $\{c_2, \ldots, c_r\} \cup D$ (each team appears $2r - 4$ times) together with an alternating orientation of the matches between team c_1 and the teams in $\{c_2, \ldots, c_r\} \cup D$ to fix the matches in a way that each team wins at most $r - 1$ matches. Finally, the remaining matches in $M_{\bar{E}}$ are fixed as follows. Team p wins all matches against teams in C and loses all matches against teams in D. All teams in V win their matches against teams in D. All teams in U lose their matches against teams in D. All teams in $T \setminus \{h, d_1, \ldots, d_{r-2}\}$ win their matches against team h and all teams in D lose their matches against team h. We complete ρ by setting the outcome probabilities of all matches in $M_E \subset M$, the matches corresponding to edges of the graph G. For each match in M_E we set the two probabilities for one of the teams to win to $1/2$.

We will now show how to construct an optimal schedule for the tournament in such a way that all open matches take place on the (last) r matchdays. Therefore, we will construct an edge coloring of the match graph in which all edges corresponding to the open matches are colored using the same r colors. The complete coloring consists of three partial colorings. We start by coloring the edges corresponding to the matches in M_E and the matches between the teams in C and $D \cup \{p, h\}$. Since the considered subgraph is an r-regular bipartite graph, it can be colored using r colors according to König's Line Coloring Theorem (see König [21]). Finding such a coloring is possible in polynomial time using the respective proof. In the second step, we consider the matches between the teams in $U \cup C$ and $V \cup D \cup \{p, h\}$ which were not considered in the previous step. Therefore, the considered subgraph is an n-regular bipartite graph which can be colored efficiently using n colors (see the previous step). Finally, we consider the matches between the teams in $U \cup C$ and the matches between the teams in $V \cup D \cup \{p, h\}$. Since the two groups have no common nodes and the number of nodes in both groups with $n + r$ teams is even, the subgraph consisting of the two cliques can be efficiently colored with $n + r - 1$ colors (see Behzad et al. [3]). Combining the three partial colorings using separate sets of colors for each of them, we receive a complete edge coloring of the graph requiring $2 \cdot (n + r) - 1$ colors which is optimal for $2 \cdot (n + r)$ nodes. As desired, the set of edges corresponding to the set of open matches M_E was colored using exactly r colors.

Finally, we return $\Phi \cdot 2^{|E|}$ as the number of perfect matchings in the bipartite graph $G = (U, V, E)$ with Φ as the answer of the evaluation oracle regarding the previously constructed instance.

To prove the validity of the reduction, we now show that the returned number $\Phi \cdot 2^{|E|}$ is the number of perfect matchings in the bipartite graph $G = (U, V, E)$. For this, we first consider the scores of the teams which result from the fixed matches. By $s_{M_{\bar{E}}}(t)$ we denote this given score for a team $t \in T$. The scores are given as follows: $s_{M_{\bar{E}}}(p) = n + r + 1$, $s_{M_{\bar{E}}}(u_i) = n + 1$ for $u_i \in U$, $s_{M_{\bar{E}}}(v_j) = n + r - 1$ for $v_j \in V$, $s_{M_{\bar{E}}}(c_i) \leq n + r$ for $c_i \in C$, $s_{M_{\bar{E}}}(d_j) \leq n + r$ for $d_j \in D$, and $s_{M_{\bar{E}}}(h) = r - 2$. For all teams except the teams in U and V, the given scores are already the final scores. We will now show that there is a one-to-one mapping between the combinations of outcomes over the set of open matches M_E in which p is the unique winner and the perfect matchings in G.

Given a perfect matching $E^* \subset E$ in G. We construct the corresponding combination of outcomes regarding the set of open matches M_E in $\mathcal{T}$ in which p is the unique winner as follows. For each edge $(u_i, v_j) \in E$ with $(u_i, v_j) \in E^*$, v_j wins the corresponding match in $\mathcal{T}$ against u_i. For each edge $(u_i, v_j) \in E$ with $(u_i, v_j) \notin E^*$, u_i wins the corresponding match in $\mathcal{T}$ against v_j. Therefore, each $v_j \in V$ wins exactly one additional match and ends up with exactly $n + r$ points. Each $u_i \in U$ wins exactly $r - 1$ additional matches and ends up with exactly $n + r$ points. Thereby p is the unique winner of the tournament with respect to the constructed combination of outcomes for the set of open matches M_E. Assume we are given a combination of outcomes with positive probability for the set of open matches M_E in which p is the unique winner. As p is the unique winner of the tournament, each $u_i \in U$ won at most $r - 1$ additional matches and each $v_j \in V$ won at most 1 additional match. As the teams in U have to lose a total of at least n matches while the teams in V have to win a total of at most n matches, the teams in V win exactly t matches in total. As each $v_j \in V$ can only win exactly one match without beating p, there exists a unique team $w(v_j) \in U$ for each $v_j \in V$ which is beaten by v_j. As the open matches in M_E correspond to the edges in G, the corresponding perfect matching in G is given by $E^* = \{(w(v_j), v_j) \mid v_j \in V\}$. This concludes the construction of the one-to-one mapping.

As each combination of outcomes has probability $(1/2)^{|E|}$, the returned number $\Phi \cdot 2^{|E|}$ corresponds to the number of perfect matchings in G. Since the reduction steps can be performed in polynomial time the #P-hardness of the problem follows. □

We now describe how the reduction can be extended to round-robin tournaments with an arbitrary number of rounds k. We fix the matches of the first $k - 1$ rounds as follows. The matches between the teams in $T \setminus \{h\}$ are fixed according to an Eulerian orientation while h loses all matches. Thereby, the score differences of the teams excluding h after the first $k - 1$ rounds remain 0. The k-th round is then constructed as given in the reduction.

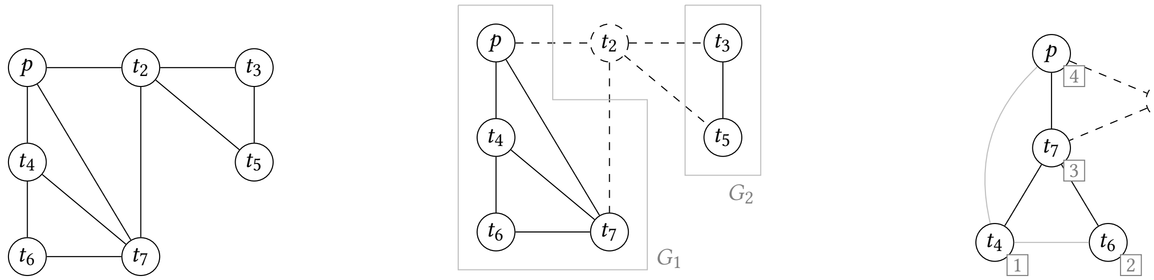
Complementary to the previous result we now consider cases in which we can achieve a certain degree of efficiency. In many cases, to determine the probability of team p being the unique winner, we do not need to examine all open matches. The outcome of an open match is only relevant to the probability if one of the two participating teams could beat team p in the end. Thus, we introduce the following concepts. Given a fixed set of outcomes O and an instance of O-Evaluation consisting of tournament $\mathcal{T} = (T, M)$, an outcome probability profile ρ regarding $\mathcal{T}$ over O, and a team $p \in T$. We call a team $t \in T \setminus \{p\}$ *critical* (for p) if there is a positive probability according to ρ that team t ends up with at least as much points as team p. Additionally, we say that p is critical by default. To check if a team $t \in T \setminus \{p\}$ is critical, we can check if team t has at least as much points as p in the case where we decide all remaining matches of p and t so that the difference in scores of the two teams is maximized in favor of t. For some cases we assume that we are additionally given a target score i for p and, in this case, a team t is critical as soon as it can reach a score of at least i. We refer to the restriction of the match graph to open matches with the participation of at least one critical team as the *critical match graph*. Furthermore, we refer to the restriction of the match graph to critical teams and open matches with the participation of exactly two critical teams as the *inner critical match graph*. A *minimum feedback arc set* (FAS) of a component of a match graph is a minimum set of edges to be removed from the multigraph of the component to make it acyclic. Note that a feedback arc set consequently contains all but one edge of each multi-edge between two team nodes. Its size is referred to as the *minimum feedback arc set size*. For undirected multigraphs it corresponds to the difference between the number of edges and the number of edges in a minimum spanning tree (MST) which is $n - 1$ for n nodes. A MST, and thus its complement a FAS, itself can be determined in polynomial time for undirected graphs using breadth-first search (BFS). We refer to the maximum minimum feedback arc set size over all components of the inner critical match graph as the *maximum fixing number* denoted by γ.

Theorem 3.4. *O-Evaluation for a fixed set of outcomes O is* FPT *when parameterized by the maximum fixing number.*

Note that here we do not require the set of outcomes O to be symmetrical, that the given tournament is a round-robin tournament, or that a schedule is given. The basic idea of the algorithm is to decompose the match graph so that a dynamic programming approach can be applied to the remaining tree structure. Figure 2 sketches the decomposition using an exemplary critical match graph.

Proof. Given a fixed set of outcomes O and an instance of O-Evaluation consisting of a tournament $\mathcal{T} = (T, M)$, an outcome probability profile ρ regarding $\mathcal{T}$ over O, and a team $p \in T$. Our algorithm for determining the probability of team p ending up as the unique winner of the tournament is given as follows.

In the first step, we calculate the teams' current scores by identifying the matches with a fixed outcome. Alternatively, one could assume that only the current scores and open matches are given



(i) Critical match graph generated from a given instance. **(ii) Identifying the components of the inner critical match graph (solid edges).** **(iii) Tree after fixing FAS matches (gray) of component G_1 with bottom-up order.**

Figure 2: Illustration of the decomposition of the match graph in the proof of Theorem 3.4.

regarding the tournament and not the entire previous history. At this point we can check if another team already has more points than p could obtain in its remaining matches. If this is the case, we can already return a probability of 0 and terminate. On the other hand, we can also check if p has already more points than any other team could obtain, in which case we could return a probability of 1.

For each possible score $i \in \{i_{min}, \dots, i_{max}\}$ of p, with i_{min} denoting the maximum fixed score over all candidates including p and i_{max} denoting the maximum score p can obtain with its remaining matches, we perform the following steps. Note that i_{min} and i_{max} are the minimum and maximum score of p for which p can have a positive probability of ending up as the unique winner.

We start by determining the critical match graph (see Figure 2 (i)) and the inner critical match graph of $\mathcal{T}$ assuming that p will achieve score i and subsequently determine the components $G_1, \dots, G_h$ of the latter one (see Figure 2 (ii)). Since p is the unique winner if all open matches with the participation of a critical team are decided in such a way that no team has at least as many points as p, we will calculate the probability for this for each component individually.

Given a component G_c of the inner critical match graph. We start by determining a minimum feedback arc set $\tilde{M_c}$ of G_c using breadth-first search to determine a minimum spanning tree and selecting the matches not contained in it.

For each combination of outputs with positive probability $\pi \in \Pi_{\tilde{M_c}}$ with $\pi : \tilde{M_c} \to O$ regarding the matches in $\tilde{M_c}$ we run the following procedure over the remaining MST of G_c (see Figure 2 (iii)).

First, update the scores according to π. Here, we can check whether a team gets i or more points by fixing the matches, and if so, ignore this combination. In bottom-up order, we traverse the teams in the tree and determine the probability $\theta_{i,t}(j)$ that team t and all teams in the underlying subtree, after deciding all open matches of those teams which are contained in the underlying subtree, will receive less than i points, the score of p, and t receives exactly j points. We can restrict j to $j \in \{j_{min}, \dots, j_{max}\}$ with j_{min} being the current score of t with respect to π and j_{max} being the minimum of i and the maximum score t can obtain in its remaining matches. If t is not p, we can also exclude i as a value for j. If j is not contained in this interval, $\theta_{i,t}(j)$ is 0 by default. If p itself is contained in G_c we assume that p is chosen as the root.

We calculate $\theta_{i,t}(j)$ using a dynamic programming approach. By $t_1, \dots, t_q$ we denote the critical teams which are children of t in the tree and uncritical teams against which t has open matches. We denote those open matches by $m_1, \dots, m_q$ respectively.

The dynamic programming table consists of the values $\theta_{i,t}(j,k)$ denoting the probability $\theta_{i,t}(j)$ restricted to the subtree only covering $t_1, \dots, t_k$ with $0 \le k \le q$. For $k = 0$ team t has only one score j_0 with positive probability which is t's score regarding the already fixed matches. Therefore, it holds that $\theta_{i,t}(j,0) = 1$ for $j = j_0$ with $j_0 < i$ and 0 otherwise for $t \neq p$ and $\theta_{i,t}(j,0) = 1$ for $j = j_0$ and 0 for $t = p$. For $k > 0$ we use the following relationship to fill out the table. For $m_k : (t, t_k)$ it holds that $\theta_{i,t}(j,k) =$

$$\sum_{(\alpha_s,\beta_s)\in O} \Big[\rho_{m_k}((\alpha_s,\beta_s)) \cdot \theta_{i,t}(j-\alpha_s, k-1) \cdot \theta_{i,t_k}(< i-\beta_s)\Big]$$

and, on the other hand, for $m_k : (t_k, t)$ it holds that $\theta_{i,t}(j,k) =$

$$\sum_{(\alpha_s,\beta_s)\in O} \Big[\rho_{m_k}((\alpha_s,\beta_s)) \cdot \theta_{i,t}(j-\beta_s, k-1) \cdot \theta_{i,t_k}(< i-\alpha_s)\Big]$$

with $\theta_{i,t_k}(< i - x)$ denoting the respective probability for t_k to receive less than $i-x$ points which is $\sum_{y=0}^{i-x-1} \theta_{i,t_k}(y)$ if t_k is a critical team and 1 if t_k is a non-critical team. Finally, it holds by definition that $\theta_{i,t}(j) = \theta_{i,t}(j,q)$ for $t \neq p$ and $\theta_{i,p}(j) = \theta_{i,p}(j,q)$ for $i = j$ and 0 otherwise. Finally, assuming that team t is the root of the tree, we set $\Phi_{i,c,\pi}$, the probability that assuming π the open matches of the teams in G_c are decided in a way that the teams in the component end up with less than i points while p receives exactly i points, to $\sum_{x=0}^{i} \theta_{i,t}(x)$. Subsequently, including all possible combinations of outcomes for the component with their respective probabilities, we set $\Phi_{i,c}$ to $\sum_{\pi\in\Pi_{\tilde{M_c}}} \rho(\pi) \cdot \Phi_{i,c,\pi}$ with $\rho(\pi) = \prod_{m\in\tilde{M_c}} \rho_m(\pi(m))$. Finally, the probability that p ends up as the unique winner of the tournament is given by $\sum_{i=i_{min}}^{i_{max}} \prod_{c=1}^{h} \Phi_{i,c}$.

All steps of the algorithm are possible in polynomial time, except the iteration over the ℓ^{γ} different combinations with constant $\ell = |O|$. Thus, the problem is fixed-parameter tractable with respect to the maximum fixing number γ for a fixed set of outcomes O. □

A further optimization for the above algorithm, which we cannot discuss in detail here, is the following. Consider a bottom-up MST for a component as described in the algorithm above. Suppose we are given the root and its children, where each child together with its subtree represents a subgraph which has been separated from the other subgraphs and the root (except for the incoming edge) by removing edges contained in the FAS. This separation allows the

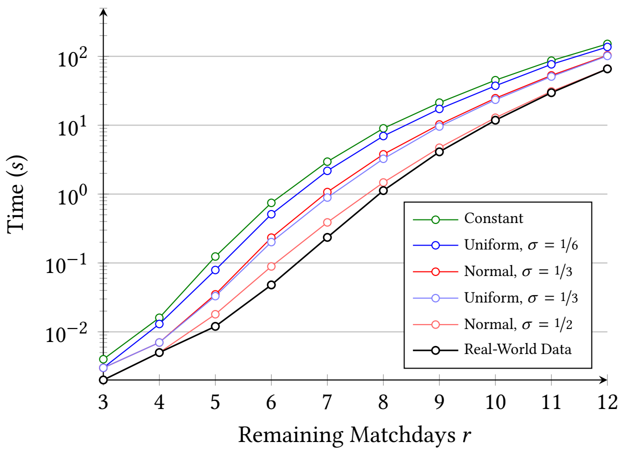


Figure 3: Average execution time on the real-world data (black) and the generated data with respect to the number of remaining matchdays.

combinations of FAS match fixations to be viewed separately for each sub-graph in each step of the recursion. Thus, the maximum number of combinations depends on the maximum number of removed edges from the original root to a leaf of the MST. We refer to the resulting new parameter as the *FAS depth*, which in the best case depends only logarithmically on the FAS size.

Consider an instance of O-RRTS-Evaluation for which only two matchdays remain open. Since each team has at most two open matches, the components of the inner critical match graph are either circles or paths. Thus, the maximum fixing number is at most one, whereby the following corollary follows by Theorem 3.4.

Corollary 3.5. *O-RRTS-Evaluation for a fixed set of outcomes O is in* FP *if there are at most two open matchdays.*

Combining Theorem 3.3 and Corollary 3.5, we obtain the dichotomy result which we announced at the beginning.

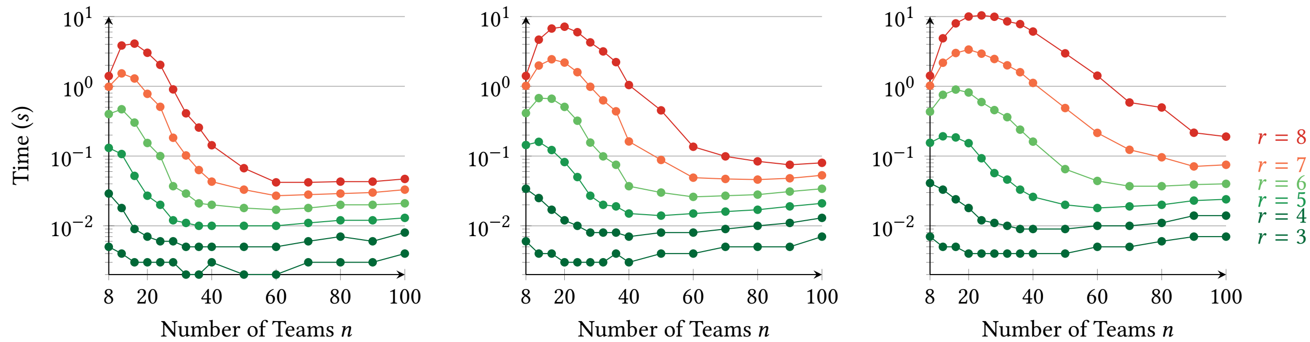
Theorem 3.6. *Given a fixed symmetric set of outcomes O, it holds that O-RRTS-Evaluation is in* FP *assuming at most two open matchdays and* #P*-hard otherwise.*

3.3 Experiments

We examine the practical relevance of the algorithm presented in the proof of Theorem 3.4 using real-world data as well as generated data. We determine the maximum execution time over the participating teams, assuming that the last r matchdays are still open. While considering the maximum over the teams instead of the average increases the times significantly, this choice is appropriate, since in most cases many teams are trivially not able to win and including them distorts the overall picture. Since it has no direct impact on the execution time, we use arbitrary probability profiles, for both the real-world and generated data, where all outcomes for all matches have a positive probability. We consider a straightforward implementation of the algorithm in Python using the `networkx` package. The experiments were performed on an Intel i5-4570t (2.90 GHz) machine with 8GB of RAM.

Real-World Data. While for many problems it is difficult to conduct comprehensive experimental analysis due to the lack of publicly available data, especially in computational social choice due to the anonymous nature of the problems, for round-robin tournaments there are vast amounts of publicly available data from different sports, countries, and leagues. We consider a total of 140 seasons from European football leagues, consisting of the participating teams, the match results, and the schedules. In detail, our dataset contains the following seasons: German Bundesliga and English Premiere League from 1996 to 2019, Spanish Primera División, Italian Serie A, and French Ligue 1 from 1998 to 2019, Portuguese Primeira Liga and Dutch Eredivisie from 2007 to 2019. From these seasons, 8 include 16 teams, 54 include 18 teams, and 78 include 20 teams. Our selection is based on the availability and integrity of the data and the use of a round-robin tournament with $k = 2$ rounds combined with the 3-point rule. We summarize the results for the real-world data and $r \in \{3, 4, \dots, 12\}$ in Figure 3.

Generated Data. In the following, we describe the data used for the performance analysis with respect to the number of teams. Note that the amount of comparable real-world data with respect to a varying number of teams is not nearly enough to perform a comprehensive analysis. We generate an instance for an arbitrary even number of teams n with $T = \{t_1, \dots, t_n\}$ as follows. As in the real-world data, we consider seasons with $k = 2$ rounds and $O = \{(3, 0), (1, 1), (0, 3)\}$. To determine the schedule of the first round, we use the so-called *circle method* also known as the canonical one-factorization, which, together with its many variations, is the most prominent method used in practice (see Goossens and Spieksma [17]). For the second round, the same schedule is used. We have also considered the variant of the circle method, where the order of the matchdays is shuffled randomly. However, there were no significant differences in the results. In order to generate plausible outcomes for the matches we use the Elo-like model presented by Ryvkin and Ortmann [26]. First, we determine an ability level profile $\vec{x} = (x_1, \dots, x_n)$ where x_i is the ability level of team t_i, by drawing them either from a normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 1/3$ or $1/2$, a continuous uniform distribution on $\left[-\sqrt{3}/6, \sqrt{3}/6\right]$ resulting in a standard deviation $\sigma = 1/6$ or on $\left[-\sqrt{3}/3, \sqrt{3}/3\right]$ resulting in a standard deviation $\sigma = 1/3$, or a constant distribution with mean 0. It is assumed that for a given match the team with the higher ability level wins, whereas the actual ability level for a match is perturbed by a normal distribution with mean 0 and standard deviation 1 with *probability density function* ϕ and *cumulative density function* Φ. Thus, the probability that team t_a beats team t_b in a match (t_a, t_b) is given by $\int_{-\infty}^{\infty} \phi(z - x_a)\Phi(z - x_b)dz$, which we evaluate numerically. Hence, we decide each match by tossing a biased coin using the calculated winning probabilities. As noted by Ryvkin and Ortmann [26], the probability for a tie in this model is 0. We also assume that an appropriate inclusion of ties here has no significant effect on the execution time. Using this method we generate a total of 500 seasons for each $n \in \{8, 12, \dots, 40, 50, \dots, 100\}$. We summarize our results for the generated data and a varying number of teams with $r \in \{3, 4, \dots, 8\}$ in Figure 4. For comparison with the real-world data in Figure 3 we generated 280 seasons with the same mixture of numbers of teams as in the real-world data.



(i) Normal distribution ($\sigma = 1/3$). **(ii) Uniform distribution ($\sigma = 1/6$).** **(iii) Constant distribution.**

Figure 4: Average execution time on the generated data depending on the ability level distribution.

Evaluation. As shown in Figure 3, the algorithm has suitable execution times for the real-world data. While the average execution time for $r \leq 7$ remaining matchdays is still well below one second, it increases to about 60 seconds for $r = 12$. We suppose that this execution time is still well justifiable for an agent with strong (financial) interest, especially if one takes into account that the experiments were performed on a rather low-powered machine and that the brute-force aspects of the algorithm can easily be parallelized. In particular, one has to consider the immense number of combinations the algorithm covers, which for $n = 16$, 18, or 20 teams and $r = 12$ open matchdays, around one-third of a season for $k = 2$ rounds, lies between $3^{96} \approx 6.363 \cdot 10^{45}$ and $3^{120} \approx 1.797 \cdot 10^{57}$, which is impossible to handle using a pure brute-force approach.

However, it cannot be denied that the performance of the algorithm develops poorly for an increasing number of open matchdays r, both for the real-world and the artificial data. Note that in the worst-case, the problem is already hard for $r = 3$. There are two main reasons for the development of the execution time with increasing r. First, the more open matchdays, the more teams have the possibility to catch up with team p, which increases the number of critical teams and second, the more open matchdays, the denser the inner critical match graph becomes. Both developments increase the maximum fixing number γ and thereby the execution time.

In the following, we will discuss the generated data in more detail. In Figure 3 we compare the average execution time of the generated data with the real-world data, where we see that all models lead to a similar behavior as for the real-world data, except for slight differences which are discussed later. Note, that we do not state here that the generated data is realistic regarding any other criteria. However, it seems to be appropriate for the analysis of the expected execution time. Depending on the setting, for example, the way the participants are selected and regulated, we can expect different distributions of ability levels. As shown in Figure 3 and 4, the distribution does have a significant effect on the complexity. The average execution time is significantly higher when the probability of top teams, with outstanding high ability levels, is lower, as for the normal or uniform distribution with lower standard deviation, and, of course, the constant distribution. This holds since the existence of a small number of significantly better teams implies that for the teams further down in the table the algorithm terminates almost directly, and second, for the top teams only a small number of critical teams exist, so the maximum fixing number γ is lower.

Theorem 3.3 suggests that, under the usual conjectures, the worst-case complexity of the problem regarding any algorithm should also increase drastically with the number of teams, even for a constant number of remaining matchdays. However, we see in Figure 4 that this is not the case in the experiments, more so, the complexity seems to decrease by several orders of magnitude after a local maximum for all three distributions. The reason for this is that even though the number of critical teams can increase, the higher the number of teams, the less likely it is that multiple matches between a set of critical teams will take place on the remaining matchdays, decreasing the likelihood of large and dense components in the inner critical match graph, and the more likely it is that a team with significantly more points than the others will emerge. The latter has already been discussed above and leads to a much faster decrease in the case of distributions with high variance, as a team with a higher ability level is more and more likely to be separated from the mass for a higher number of matches.

4 CONCLUSIONS

We examined the parameterized complexity of scheduling and predicting round-robin tournaments regarding the number of prescheduled and remaining matchdays. Beside the theoretical hardness of predicting the outcomes of those tournaments, we presented, to the best of our knowledge, the first exact algorithm with suitable execution time for real-world application.

The possibilities for future work are numerous. In addition to the consideration of related problems, approximation, and the average-case complexity instead of the worst-case complexity, the approaches presented here can be further developed. For example, by extending and adapting the algorithm and the previous definitions, it is possible to predict other events such as qualification and relegation and to consider tie-breaking mechanisms.

ACKNOWLEDGMENTS

This work is supported by the DFG-grant BA6270/1-1.

REFERENCES

[1] S. Arora and B. Barak. 2009. *Computational Complexity: A Modern Approach*. Cambridge University Press.

[2] D. Baumeister and T. Hogrebe. 2020. Complexity of Election Evaluation and Probabilistic Robustness (Extended Abstract). In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*. 1771–1773.

[3] M. Behzad, G. Chartrand, and J. Cooper. 1967. The colour numbers of complete graphs. *Journal of the London Mathematical Society* 1, 1 (1967), 226–228.

[4] T. Bernholt, A. Gülich, T. Hofmeister, and N. Schmitt. 1999. Football Elimination is Hard to Decide Under the 3–Point–Rule. In *International Symposium on Mathematical Foundations of Computer Science*. Springer, 410–418.

[5] K. Cechlárová, E. Potpinková, and I. Schlotter. 2016. Refining the complexity of the sports elimination problem. *Discrete Applied Mathematics* 199 (2016), 172–186.

[6] A. Chetwynd and A. Hilton. 1985. Regular Graphs of High Degree are 1-Factorizable. *Proceedings of the London Mathematical Society* 3, 2 (1985), 193–206.

[7] W. Cohen, R. Schapire, and Y. Singer. 1998. Learning to Order Things. In *Advances in Neural Information Processing Systems*. 451–457.

[8] C. Colbourn. 1983. Embedding Partial Steiner Triple Systems Is NP-Complete. *Journal of Combinatorial Theory, Series A* 35, 1 (1983), 100–105.

[9] B. Csaba, D. Kühn, A. Lo, D. Osthus, and A. Treglown. 2016. *Proof of the 1-factorization and Hamilton decomposition conjectures*. Vol. 244. American Mathematical Society.

[10] P. Dagum and M. Luby. 1992. Approximating the permanent of graphs with large factors. *Theoretical Computer Science* 102, 2 (1992), 283–305.

[11] G. Dirac. 1952. Some Theorems on Abstract Graphs. *Proceedings of the London Mathematical Society* 3, 1 (1952), 69–81.

[12] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. 2001. Rank Aggregation Methods for the Web. In *Proceedings of the 10th International Conference on World Wide Web*. ACM, 613–622.

[13] E. Ephrati and J. Rosenschein. 1993. Multi-Agent Planning as a Dynamic Search for Social Consensus. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*. 423–429.

[14] M. Faella and L. Sauro. 2021. Irrelevant matches in round-robin tournaments. *Autonomous Agents and Multi-Agent Systems* 35, 1 (2021), 1–34.

[15] H. Fleischner. 1991. Algorithms for Eulerian Trails. In *Eulerian Graphs and Related Topics: Part 1 (Annals of Discrete Mathematics)*. Vol. 2. Chapter X.1.

[16] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. 1999. Voting for movies: the anatomy of a recommender system. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*. ACM, 434–435.

[17] D. Goossens and F. Spieksma. 2012. Soccer schedules in Europe: an overview. *Journal of Scheduling* 15, 5 (2012), 641–651.

[18] N. Hazon, Y. Aumann, S. Kraus, and M. Wooldridge. 2012. On the evaluation of election outcomes under uncertainty. *Artificial Intelligence* 189 (2012), 1–18.

[19] I. Holyer. 1981. The NP-Completeness of Edge-Coloring. *SIAM J. Comput.* 10, 4 (1981), 718–720.

[20] W. Kern and D. Paulusma. 2004. The computational complexity of the elimination problem in generalized sports competitions. *Discrete Optimization* 1, 2 (2004), 205–214.

[21] D. König. 1916. Graphok és alkalmazásuk a determinánsok és a halmazok elméletére. *Mathematikai és Természettudományi Ertesito* 34 (1916), 104–119.

[22] N. Mattei, J. Goldsmith, A. Klapper, and M. Mundhenk. 2015. On the complexity of bribery and manipulation in tournaments with uncertain information. *Journal of Applied Logic* 13, 4 (2015), 557–581.

[23] C. Papadimitriou. 1994. *Computational Complexity*. Addison-Wesley.

[24] L. Robinson. 1991. Baseball playoff eliminations: An application of linear programming. *Operations Research Letters* 10, 2 (1991), 67–74.

[25] A. Rosa and W. Wallis. 1982. Premature Sets of 1-Factors or How Not to Schedule Round Robin Tournaments. *Discrete Applied Mathematics* 4, 4 (1982), 291–297.

[26] D. Ryvkin and A. Ortmann. 2008. The Predictive Power of Three Prominent Tournament Formats. *Management Science* 54, 3 (2008), 492–504.

[27] S. Saarinen, J. Goldsmith, and C. Tovey. 2015. Probabilistic Copeland Tournaments. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. 1851–1852.

[28] P. Scarf and X. Shi. 2008. The importance of a match in a tournament. *Computers & Operations Research* 35, 7 (2008), 2406–2418.

[29] B. Schwartz. 1966. Possible Winners in Partially Completed Tournaments. *SIAM Rev.* 8, 3 (1966), 302–308.

[30] M. Spann and B. Skiera. 2009. Sports Forecasting: A Comparison of the Forecast Accuracy of Prediction Markets, Betting Odds and Tipsters. *Journal of Forecasting* 28, 1 (2009), 55–72.

[31] E. Zermelo. 1929. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift* 29, 1 (1929), 436–460.