Operations Research Letters 44 (2016) 96–100

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Scheduling asynchronous round-robin tournaments

Warut Suksompong

Department of Computer Science, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:
Received 25 September 2015
Received in revised form
1 December 2015
Accepted 1 December 2015
Available online 9 December 2015

Keywords:
Timetabling
Sports scheduling
Round-robin tournament
Fairness

ABSTRACT

We study the problem of scheduling asynchronous round-robin tournaments. We consider three measures of a schedule that concern the quality and fairness of a tournament. We show that the schedule generated by the well-known "circle design" performs well with respect to all three measures when the number of teams is even, but not when the number of teams is odd. We propose a different schedule that performs optimally with respect to all measures when the number of teams is odd.

1. Introduction

A round-robin tournament, also known as an all-play-all tournament, is a popular format for organizing sports competitions. In a round-robin tournament, every pair of teams play each other a fixed number of times during the competition. Since every team competes with every other team, the winner of a round-robin tournament is usually thought to depend much less on luck than that of, say, a knockout tournament. A series of work has investigated how to schedule a round-robin tournament when different notions are central to the organizers' consideration. One line of research has focused on time-relaxed tournaments, which takes into account the issue of time off between games involving the same team [7,8,11], while another has considered fairness issues [3,4,12,14]. We refer the interested reader to a survey by Rasmussen and Trick [9] and a book by Anderson [1] for more detail on the literature.

In this paper, we study the problem of scheduling asynchronous round-robin tournaments, i.e., round-robin tournaments in which no two games take place at the same time. There are a number of reasons why it might be desirable to schedule all games at different times. Indeed, this tournament format allows spectators to follow all the games live, and the organizers can maximize revenue while having to organize the same number of games. Tournaments may even need to be asynchronous if there is only one venue where a game can take place. An example of an asynchronous round-robin tournament is the 2012 Premier League Snooker in England, in which five players in the group stage play a total of ten games in ten different weeks (albeit in ten different venues as well).

When scheduling an asynchronous round-robin tournament, the organizers may desire properties that improve the quality and fairness of the tournament. Unlike in knockout tournaments, where the organizers can significantly impact the outcome of the tournament by setting up a bracket of their choice (see, e.g., [6,13]), the set of games to be played in a round-robin tournament cannot be changed. Nevertheless, the order in which the games are played can still be an important factor in a round-robin tournament. For example, when teams have a longer rest between games, they are more likely to have a relaxing rest and perform at their full potential in the next game. On the other hand, if some team has a long rest going into a game while its opponent has just played its previous game, the former team could be at a clear advantage. Another desirable property of a schedule is that at any point during the tournament, all teams should have played roughly the same number of games. This prevents the advantage of knowing too many results involving other teams and the possibility of collusion as well. We define measures that capture all of these properties, and we exhibit schedules that perform (close to) optimally with regard to our measures. In particular, we show that the schedule generated by the well-known "circle design" performs well with respect to all three measures when the number of teams is even, but not so well when the number of teams is odd. We also propose a different schedule that performs optimally with respect to all three measures when the number of teams is odd. We hope that this schedule will be of practical interest to organizers of asynchronous round-robin tournaments.

A related problem that is worth mentioning is the problem of finding balanced tournament designs, which has been considered

E-mail address: warut@cs.stanford.edu.

http://dx.doi.org/10.1016/j.orl.2015.12.008
0167-6377/© 2015 Elsevier B.V. All rights reserved.

Table 1
Summary of our results. All bounds are known to be attainable except that for the rest difference index when n is even. See also Section 5 for further discussion.

	Circle method, n even	Any schedule, n even	Circle method, n odd	Any schedule, n odd
Guaranteed rest time	$(n-4)/2$	$\leqslant (n-4)/2$	$(n-5)/2$	$\leqslant (n-3)/2$
Games-played difference index	1	$\geqslant 1$	2	$\geqslant 1$
Rest difference index	1 if $n=4$; 2 if $n \geqslant 6$	$\geqslant 1$	$(n+1)/2$	$\geqslant 1$

by some prior work [2,5,10]. In the setting of balanced tournament designs, it is assumed that there exist external factors that make some games different from others, and it is desirable that teams receive roughly the same effect from these external factors. For instance, the tournament might involve games during different times of the day or at different venues. Since some teams might be more familiar with playing in the morning than in the evening or with playing at one venue than another, the aim of a balanced tournament design is to eliminate or minimize the potential advantage by scheduling teams to play as evenly across the different times and venues as possible. On the other hand, in our setting there is no inherent difference between games. Indeed, a good example to keep in mind throughout this paper is that the games in the tournament are scheduled on consecutive days, one game per day, at a single venue.

A summary of our results can be found in Table 1.

2. Preliminaries

We assume that the tournament in consideration is a single round-robin tournament, i.e., every pair of teams play each other exactly once. As we will mention in Section 5, however, several of our results can be generalized to arbitrary round-robin tournaments as well.

Let n denote the number of teams in the tournament. We divide the games in the tournament into r rounds of g games, where the first round comprises the first g games, the second round the next g games, and so on. The parameters r and g depend on n and are given by

$$g = \left\lfloor \frac{n}{2} \right\rfloor,$$

$$r = 2 \cdot \left\lceil \frac{n}{2} \right\rceil - 1 = \begin{cases} n & \text{if } n \text{ is odd;} \\ n-1 & \text{if } n \text{ is even.} \end{cases}$$

A team is said to play in slot i in a round if it plays the ith game of that round. We emphasize that in asynchronous tournaments, rounds do not carry any particular meaning in the implementation of the tournament and are defined merely for the sake of convenience of our analysis.

A single round-robin tournament consists of $\binom{n}{2} = \frac{n(n-1)}{2}$ games. Each team plays $n-1$ games, and we have the identity

$$r \cdot g = \left(2 \cdot \left\lceil \frac{n}{2} \right\rceil - 1\right) \cdot \left\lfloor \frac{n}{2} \right\rfloor = \frac{n(n-1)}{2}.$$

A well-known method for scheduling a round-robin tournament, described for instance by Haselgrove and Leech [5], is called the *circle design*. The method works as follows. Assume first that n is even. We arrange the teams into two rows of $n/2$ teams in such a way that the two rows align team by team. The games in the first round correspond to the pairs of teams that are aligned in this arrangement. For asynchronous tournaments, we read the games from left to right. To generate the games in the next round, we keep the top-left team fixed and rotate the remaining teams one step counterclockwise. (It is also possible to rotate the remaining teams one step *clockwise*, but this results in the same schedule as rotating counterclockwise under appropriate renaming of the teams.) We perform the rotation $n-2$ times to generate the games in all $n-1$ rounds. If n is odd, we simply pretend that the top-left team is a dummy team, and whichever team is matched to that team "sits out" the round (i.e., gets a bye in that round). The first three rounds for the tournaments with $n=10$ and $n=11$ are shown in Figs. 1 and 2, respectively.

1	2	3	4	5		1	10	2	3	4		1	9	10	2	3
10	9	8	7	6		9	8	7	6	5		8	7	6	5	4

Fig. 1. The first three rounds generated by the circle design for a tournament with $n=10$.

1	2	3	4	5		11	1	2	3	4		10	11	1	2	3
10	9	8	7	6		9	8	7	6	5		8	7	6	5	4

Fig. 2. The first three rounds generated by the circle design for a tournament with $n=11$. Note that one team "sits out" each round (i.e., gets a bye in that round).

We now define three measures of a schedule for an asynchronous tournament that concern the quality and fairness of the tournament. The first measure, guaranteed rest time, considers the minimum amount of time that the schedule allows teams to take a rest before their next game.

Definition 2.1. The *guaranteed rest time* of a schedule for an asynchronous tournament is the maximum integer b such that in the schedule, any two games involving a team is separated by at least b games not involving that team.

A schedule with a high guaranteed rest time is desirable, as it allows teams to take a long rest and prepare themselves for the next game. The higher the guaranteed rest time, the more likely we will see teams perform at their full potential in the tournament.

The next two measures, the games-played difference index and the rest difference index, reflect the fairness of the schedule.

Definition 2.2. The *games-played difference index* of a schedule for an asynchronous tournament is the minimum integer p such that at any point in the schedule, the difference between the number of games played by any two teams is at most p.

It is evident that for any tournament with at least three teams, the games-played difference index is at least 1. A schedule with a low games-played difference index ensures that all teams have played roughly the same number of games at any point during the tournament. This prevents the advantage that some teams may have if they know the results of too many games involving other teams. Indeed, with this knowledge the teams can adjust their strategy to achieve their desired position in the tournament and may even conspire with one another to do so.

Definition 2.3. The *rest difference index* of a schedule for an asynchronous tournament is the minimum integer d such that for any game in the schedule, if one team has not played in i_1 consecutive games since its last game and the other team has not played in i_2 consecutive games since its last game, then $|i_1 - i_2| \leqslant d$. (To handle the situation in which a team is playing its first game in the tournament, we will assume that all teams are involved in an imaginary game that takes place one slot before the first game of the schedule.)

It is again evident that for any tournament with at least three teams, the rest difference index is at least 1. A schedule with a low rest difference index guarantees that the two teams involved in a game have approximately the same amount of rest time going into the game.