Asynchronous Round-Robin Tournaments

Rolf Sander

http://www.rolf-sander.net/software/asyroro.html

Motivation

Imagine you are organizing a one-day football tournament. Five teams will come, every team should play every other team, and you only have one field. This is called an "asynchronous round-robin tournament". With itertools, python makes it easy to print a list of all games:

```
import itertools
list(itertools.combinations(range(5), 2))
```

The output will be:

[(0,1), (0,2), (0,3), (0,4), (1,2), (1,3), (1,4), (2,3), (2,4), (3,4)]

Here, the first element (0,1) means that team 0 plays against team 1 in the first game, the second element (0,2) means that team 0 plays against team 2 in the second game, and so on. Unfortunately, the order of games is not suitable. Team 0 has to play four games in a row without a rest, and most other teams also have to play two or more consecutive games. In contrast, asyroro.py creates a well-balanced schedule.

Usage

Simply type "python asyroro.py" to run a few examples. Adjust the code to your own needs. For a tournament with 5 teams (called A, B, C, D, E here), the 10 games of the calculated schedule are:

(A-B) (D-C) (E-A) (B-C) (D-E) (C-A) (B-D) (C-E) (A-D) (E-B)

Implementation

For an even number of teams, the circle method [1, 2] is used to generate the schedule. For an odd number of teams, the circle method does not produce satisfactory results, and the method suggested by Suksompong [3] is used instead. To balance the schedule according to the fourth criterion below, a method similar to that for creating a regular tournament presented by Herke [4] is used.

Quality control

Several criteria define the quality of a well-balanced schedule:

- **Guaranteed Rest Time (GRT)**
  As already mentioned above, each team should get a rest before they have to play again. The minimum rest for any team is the "guaranteed rest time" GRT [3]. If, for example, the schedule has a GRT of 2, then every team will have a break of at least 2 games before they have to play again.

  For asyroro-generated schedules, GRT = k-2 for an even number of teams (2*k) and GRT = k-1 for an odd number of teams (2*k+1).

- **Games-Played Difference (GPD)**
  At any time during the tournament, each team should already have played a similar number of games. The games-played difference GPD [3] is the largest such difference that occurs in the schedule.

  For all asyroro-generated schedules, GPD = 1

- **Rest Difference Index (RDI)**
  When two teams play each other, they had a different rest time before that game (note that if they had the same rest time, they would have already played against each other in the previous game). The maximum rest difference that occurs in the schedule is called the rest difference index RDI [3].

  For asyroro-generated schedules, RDI = 2 for an even number of teams and RDI = 1 for an odd number of teams.

- **Tournament Irregularity Index (TII)**
  The two halves of the playing field may not have the same quality, for example on one side the players would have to look directly into the setting sun, which is a disadvantage. Therefore, all teams should have roughly the same number of games in each half. Let's say the notation (0-1) means that team 0 plays on the left side and team 1 on the right side. If the schedule for a tournament with 3 teams is (0-1) (0-2) (1-2), then the second game needs to be changed in order to get a balanced schedule: (0-1) (2-0) (1-2). In graph theory, such a balanced (or almost balanced) schedule is called a regular (or semi-regular) tournament. Here, I call the maximum difference between games on the left side and games on the right size for any team the tournament irregularity index TII.

  For asyroro-generated schedules, TII = 1 (semi-regular) for an even number of teams and TII = 0 (regular) for an odd number of teams.

References

[1] Arunachalam Y. Tournament scheduling. `https://nrich.maths.org/1443`.

[2] Wikipedia. Round-robin tournament. `https://en.wikipedia.org/wiki/Round-robin_tournament#Scheduling_algorithm`.

[3] W. Suksompong. Scheduling asynchronous round-robin tournaments, 2016. `http://dx.doi.org/10.1016/j.orl.2015.12.008`.

[4] S. Herke. Fun with graphs, 2014. `https://www.youtube.com/watch?v=tKpariULmPI`.

This document was last changed: 2017-06-15