Minimal and fair waiting times for single-day sports tournaments with multiple fields

Lisa Garcia Tercero[1,2] [0009−0005−9058−0508], Dries Goossens[1,3] [0000−0003−0224−3412], and David Van Bulck[1,3] [0000−0002−1222−4541]

[1] Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium
{lisa.garciatercero,dries.goossens,david.vanbulck}@ugent.be
[2] KU Leuven, Department of Computer Science, CODeS, Gebroeders De Smetstraat 1, 9000 Ghent, Belgium
[3] FlandersMake@UGent - core lab CVAMO

Keywords: Sports timetabling, Integer programming, Fairness

1 Introduction

In this extended abstract, we consider a sports tournament where all participants gather at one location for a day and a limited number of fields is available. We focus on a single round-robin tournament (i.e. all teams have to play each other exactly once), where each team needs a resting time of at least one time slot between every two games they play. This setting occurs regularly in practice; our work is particularly inspired by an amateur badminton tournament called “PK WVBF” in West-Flanders, Belgium. Sometimes having more resting time is perceived as desirable, yet in amateur tournament environments teams prefer to play their games in quick succession so they can return home without delay. Therefore, we generate timetables minimizing waiting times, defined for each team as the total number of time slots they have to be present in addition to their games and resting times. We focus on creating timetables that are both efficient (by minimizing the total waiting time) and fair (by minimizing the maximum waiting time).

Knust [2] considers a tournament which is similar except that it spans multiple days and only one field is available. Moreover, the tournament is divided into different blocks (or playing days) such that each team plays twice per block. Knust proposes an exact polynomial-time algorithm that simultaneously optimizes the total and maximum waiting times. We will generalize Knust’s algorithm into a heuristic to generate timetables for a single-day tournament with multiple fields. Note that the single-day constraint complicates the timetabling efforts, since the blocks created by Knust’s algorithm cannot simply be scheduled one after the other due to the required resting times.

For reviews on generating timetables for sports tournaments, we refer to the surveys [1], [3], and [4].

2 Problem formulation

In the IP model we consider the number of time slots and fields as a given. First, given a set of teams T, we assume the number of fields equals $\lceil (|T|-1)/4 \rceil$. Considering only $\lceil (|T|-1)/2 \rceil$ games can be played simultaneously and a team can only play one game per two time slots, increasing the number of fields would mainly increase the number of empty slots without offering much advantage in terms of our objectives. With this number of fields, preliminary experiments show that timetables can be created of length $2|T|$. Furthermore, we assume a day is always long enough to schedule all games. We assign a time slot to each game, such that (i) the number of fields is always respected, (ii) the required resting times are respected, and (iii) the total waiting time and the maximum waiting time are minimized.

Parameters

T	Set of teams
G	Set of games, constituting a single round-robin tournament
$G_t \subset G$	Set of games that team $t \in T$ has to play
S	Set of time slots, $\|S\| = 2\|T\|$
f	Number of fields, $f = \lceil (\|T\|-1)/4 \rceil$

Decision variables

x_{gs}	1 if game $g \in G$ is scheduled at time slot $s \in S$, 0 otherwise.
w_t	waiting time of team $t \in T$

The model is formulated as follows, using a lexicographic bi-objective function:

$$\min \sum_{t \in T} w_t \tag{1}$$

$$\min \left(\max_{t \in T} w_t \right) \tag{2}$$

subject to

$$\sum_{s \in S} x_{gs} = 1 \qquad \forall g \in G \tag{3}$$

$$\sum_{g \in G} x_{gs} \leq f \qquad \forall s \in S \tag{4}$$

$$\sum_{g \in G_t} x_{gs} + x_{g(s+1)} \leq 1 \qquad \forall t \in T, s = 1, ..., |S|-1 \tag{5}$$

$$\sum_{s \in S} \left((s+1)x_{gs} - s x_{g's} \right) \leq w_t + (2|T|-1) \qquad \forall t \in T, \forall g, g' \in G_t \tag{6}$$

$$x_{gs} \in \{0, 1\} \qquad \forall g \in G, s \in S \tag{7}$$

$$w_t \geq 0 \qquad \forall t \in T \tag{8}$$

The first objective minimizes the total waiting time, whereas the second minimizes the maximum waiting time across all teams. Constraints (3) and (4) ensure that every game is played exactly once and that games are restricted to the number of available fields.

Furthermore, Constraints (5) guarantee that the required resting times are respected. Next, the waiting time of each team is defined by Constraints (6); the games and resting times of each team require $2|T|-1$ time slots. Finally, Constraints (7) and (8) are variable domain constraints. Note that the integrality of w_t follows from Constraints (6).

3 Preliminary computational results

In this section, we show how to generalize Knust's algorithm [2] into a heuristic for a single-day tournament with multiple fields. The strategy is to generate blocks such that each team plays two games per block, and schedule them consecutively on a single day. In order to do this, we assume $|T|$ to be odd as this enables us to follow the structure of the timetables generated by Knust. The blocks are constructed by generating the games in the same order as the single-field variant and scheduling them at the earliest feasible time slot, resulting in a total waiting time of 2 per block. Intuitively, this follows from the fact that some games cannot be scheduled on their “ideal” time slot due to all fields already being occupied. We ensure resting times are respected by scheduling an empty slot between the blocks, after which games are moved forward individually. Details will be discussed during the talk.

Since the number of fields equals $\lceil(|T|-1)/4\rceil$ and $|T|$ games are played per block, we need at least 4 slots per block; the heuristic creates blocks of length 5. Together with the empty slots after each block and the fact that $(|T|-1)/2$ blocks have to be planned, we have an upper bound on the number of time slots of $3(|T|-1)$.

We compare the running time of the heuristic to the IP model, solved with Gurobi version 10.0.03, in Table 1.

Table 1: Running times of the IP model and the heuristic for odd numbers of teams.

|T|	IP model	Heuristic
11	2m46s	0s
13	1m12s	12ms
15	8m38s	1ms
17	>30m	1ms
...	...	...
501		2.8s

Despite the IP model being too slow to create optimal schedules for a large number of teams, we suspect that for each odd $|T|$ a schedule exists such that:

$$\sum_{t\in T} w_t = 2|T| - 4 \tag{9}$$

$$\max_{t\in T} w_t = 2 \tag{10}$$

This conjecture is confirmed for the cases where $|T| \leq 15$. Assuming these results hold for every odd $|T|$, we compare them against our heuristic in Figure 1.

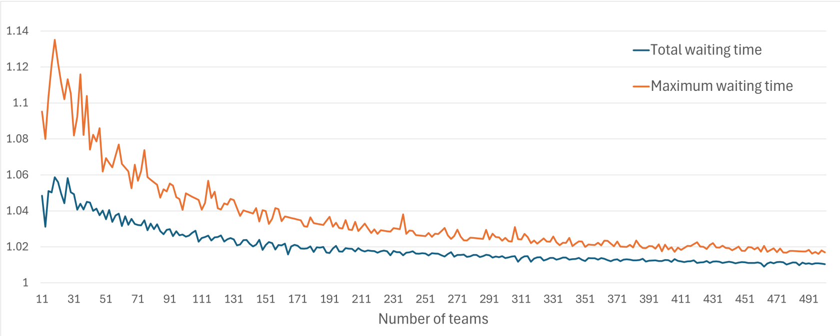

Fig. 1: Ratio of the results obtained by the heuristic to the conjectured optimal results, for odd numbers of teams $|T|$.

This extended abstract represents an initial step towards the general problem of a single-day round-robin tournament with any given number of available fields and any number of teams participating. For instances where the number of teams is even, we can easily add a dummy team. However, the impact of this on waiting times requires investigation. Furthermore, we plan to examine how the structure of the schedules should be adapted when fewer fields are available, while keeping the waiting times within limits.

References

1. Kendall, G., Knust, S., Ribeiro, C., Urrutia, S.: Scheduling in sports: An annotated bibliography. Computers & Operations Research **37**(1), 1–19 (2010)
2. Knust, S.: Scheduling sports tournaments on a single court minimizing waiting times. Operations Research Letters **36**(4), 471–476 (2008)
3. Ribeiro, C.: Sports scheduling: Problems and applications. International Transactions in Operational Research **19**(1-2), 201–226 (2012)
4. Van Bulck, D., Goossens, D., Schönberger, J., Guajardo, M.: RobinX: A three-field classification and unified data format for round-robin sports timetabling. European Journal of Operational Research **280**(2), 568–580 (2020)