Generalizing first-break-then-schedule to time-relaxed sports timetabling

David Van Bulck · Dries Goossens

Received: date / Accepted: date

Abstract A popular technique to construct time-constrained timetables is the so-called first-break-then-schedule approach which first determines for each team the time slots on which it plays home or away, after which the opponent of each time slot is determined. Whereas in time-constrained timetables the number of time-slots is just enough to play all games, time-relaxed timetables utilize more time slots than there are games per team. This offers time-relaxed timetables additional flexibility to take into account venue and player availability constraints. Despite their flexibility, time-relaxed timetables have the drawback that the rest period between teams' consecutive games can vary considerably, the difference in rest time between opponents may become unequal, and the difference in the number of games played at any point in the season can become large. In this paper, we explore how to generalize techniques based on first-break-then-schedule to generate time-relaxed timetables that are less prone to these drawbacks.

Keywords Time-relaxed sports timetabling · Availability constraints · Pattern decomposition · First-break-then-schedule · First-off-day-then-schedule · Game-off-day pattern set feasibility

1 Introduction

Every sports competition needs a timetable, also called schedule, which defines who will play whom. We assume that each game has two opponents, and that a game is played in the venue of the home team (the other team plays

D. Van Bulck
Ghent University, Faculty of Economics and Business Administration
E-mail: david.vanbulck@ugent.be

D. Goossens
Ghent University, Faculty of Economics and Business Administration
E-mail: dries.goossens@ugent.be

away). This paper studies the construction of a timetable for a so-called time-relaxed double round-robin tournament (for an overview on sports timetabling, see Van Bulck et al. (2020)). In a double round-robin tournament (2RR), each teams meets every other team once at home, and once away. A closely related variant of the 2RR is the single round-robin tournament (1RR) where each team meets every team exactly once. In contrast to time-constrained timetables, time-relaxed timetables utilize more time slots than minimally needed to schedule all the games. This allows time-relaxed timetables to take into account venue availability constraints that state when a team can play home, and team unavailability constraints that state when a team cannot play at all.

Their flexibility notwithstanding, time-relaxed timetables face three fairness issues that do not occur in time-constrained timetabling. First, the difference in the number of games played per team after each time slot causes tournament rankings to become inaccurate. The *games-played difference index* of a timetable is 'the minimum integer GPDI such that at any point in the schedule, the difference between the number of games played by any two teams is at most GPDI (Suksompong, 2016)'. Second, the rest time between teams' consecutive games can vary substantially, which can result in congested periods or long periods without any game. Since several researchers found a relation between fixture congestion and higher injury rates (e.g. Bengtsson et al. (2013), Dupont et al. (2010)), a popular constraint is to limit the maximal allowed sequence of games and off days. Third, the difference in rest time between two opponents may give an advantage to the most rested team. The *rest difference index* equals 'the minimum integer RDI such that for any game in the timetable, 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 $|i_1 - i_2| \leqslant$ RDI (Suksompong, 2016)'.

Observing that the difference in rest time is only relevant if the least rested team has not fully recovered from its previous game, this paper proposes to minimize the *sum of truncated rest differences* while at the same time controlling for absolute rest time and difference in games played. The truncated rest time of a team equals its absolute rest time r if $r < \tau$ and equals τ otherwise. For a more profound empirical motivation of this truncation, we refer to Scoppa (2015). Without truncation (i.e., $\tau = \infty$), the objective proposed in this paper is known in the literature as the total rest difference objective (see Çavdaroğlu and Atan (2020)) and as the rest mismatch objective if in addition the magnitude of the difference is ignored (see Atan and Çavdaroğlu (2018)).

Due to the large number and diversity of constraints and objectives that typically need to be considered, the construction of sports timetables is challenging. This difficulty has led to a wide variety of sports timetabling approaches. Many of these methods have in common that they decompose the problem into different subproblems, but differ in the order the subproblems are solved and the methods chosen for each subproblem (e.g., see Trick (2001)). One particular decomposition method is *first-break-then-schedule* which first fixes for each time slot which teams play home and which play away, after which it determines the opponents in each time slot (see Section 3). First-break-then-schedule has been used to successfully schedule a wide variety of time-constrained competitions, including the Atlantic Coast Conference (Nemhauser and Trick, 1998), the Danish football league (Rasmussen, 2008), and the Belgian football league (Goossens and Spieksma, 2009). Despite its popularity, first-break-then-schedule has not been used so far to schedule time-relaxed competitions. This paper investigates whether first-break-then-schedule can be generalized so as to deal with the above described fairness issues in time-relaxed timetables.

2 Problem description and integer programming formulation

In the time-relaxed availability constrained double round-robin tournament described below, the input consists of a set of arbitrarily many time slots S, a set of teams T with $|T| = n$, for each team $i \in T$ a player availability set $A_i \subseteq S$ and venue availability set $H_i \subseteq A_i$, and four integers GPDI, ρ, σ, and τ. A feasible timetable for this problem assigns each game (i, j) of the double round-robin tournament, with home team $i \in T$ and away team $j \in T \setminus \{i\}$, to a time slot $s \in S$ such that each team plays at most once per time slot and:

(C1) the player availability A_i is respected for all teams (i.e., no game (i, j) or (j, i) is planned on a time slot $s \notin A_i$),
(C2) the venue availability H_i is respected for the home teams (i.e., no game (i, j) is planned on a time slot $s \notin H_i$),
(C3) the games played difference index is at most GPDI,
(C4) each team plays at most two games per ρ consecutive time slots, and
(C5) each team has at most σ consecutive off days.

In addition, the objective is to minimize the sum of truncated rest differences, thereby assuming that teams are fully recovered from their previous game after τ time slots and that they are fully rested at the start of the season.

Availability constraints *(C1)* and *(C2)* have applications in a multitude of time-relaxed tournaments. For example, non-professional teams typically share their venues with other teams and their players need to be able to combine their sport with work and family (e.g., Schönberger et al. (2004), Van Bulck et al. (2019)). A timetable with a low maximal difference in games played *(C3)* is desirable since this results in more accurate tournament rankings and may reduce the opportunities for match fixing. In a 2RR, the games-played difference index may be as high as $2(n-2)$: one team has played twice against every other team except for one team that did not play any game yet. Constraints *(C4)* and *(C5)* respectively limit the maximal and minimal number of games in a series of consecutive time slots so as to avoid fixture congestion and long periods without any game. Since teams might blame the timetable for losing the game if they have less rest than their opponent, the objective is to minimize the sum of truncated rest differences.

Without the objective function and Constraints *(C3)* to *(C5)* (i.e. GPDI $= 2(n-2)$, $\rho = 2$, and $\sigma = |S| - 2(n-1)$), the problem is known in the literature as 'RAC-2RR'. From a theoretical point of view, Van Bulck and Goossens (2020a) show that RAC-2RR is $\mathcal{NP}$-complete.

Equations (1)-(17) present an integer programming (IP) formulation for the problem just described. Our main decision variable is $x_{i,j,s}$, which is 1 if team $i \in T$ plays a home game against team $j \in T \setminus \{i\}$ in time slot $s \in S$, and 0 otherwise. Variable $q_{i,s}$ represents the number of games played by team i up to and including time slot $s \in S$, and variable $y_{i,s,t}$ is 1 if team i plays a game in time slot s, followed by its next game in time slot t, for each $s, t \in S$ such that $s < t \leqslant \tau + s$, and 0 otherwise. Finally, variable $d_{i,j}$ contains the truncated rest difference of game (i, j).

$$\textbf{minimize} \sum_{i,j \in T: i \neq j} d_{i,j} \tag{1}$$

subject to

$$\sum_{s \in H_i \cap A_j} x_{i,j,s} = 1 \qquad \forall i, j \in T : i \neq j \tag{2}$$

$$\sum_{j \in T \setminus \{i\}} (x_{i,j,s} + x_{j,i,s}) \leqslant 1 \qquad \forall i \in T, \forall s \in S \tag{3}$$

$$q_{i,1} = \sum_{j \in T \setminus \{i\}} (x_{i,j,1} + x_{j,i,1}) \qquad \forall i \in T \tag{4}$$

$$q_{i,s} = q_{i,s-1} + \sum_{j \in T \setminus \{i\}} (x_{i,j,s} + x_{j,i,s}) \qquad \forall i \in T, s \in S \setminus \{1\} \tag{5}$$

$$q_{i,s} - q_{j,s} \leqslant \text{GPDI} \qquad \forall i, j \in T : i \neq j, \forall s \in S \tag{6}$$

$$\sum_{j \in T \setminus \{i\}} \sum_{p=s}^{s+\rho-1} (x_{i,j,p} + x_{j,i,p}) \leqslant 2 \qquad \forall i \in T, \forall s \in S : s + \rho - 1 \leq |S| \tag{7}$$

$$\sum_{j \in T \setminus \{i\}} \sum_{p=s}^{s+\sigma} (x_{i,j,p} + x_{j,i,p}) \geqslant 1 \qquad \forall i \in T, \forall s \in S : s + \sigma \leq |S| \tag{8}$$

$$\sum_{j \in T \setminus \{i\}} \Big(x_{i,j,s} + x_{j,i,s} + x_{i,j,t} + x_{j,i,t} - \sum_{k=s+1}^{t-1} (x_{i,j,k} - x_{j,i,k}) \Big) - 1 \leqslant y_{i,s,t} \qquad \forall i \in T, s, t \in S : s < t \leqslant \tau + s \tag{9}$$

$$y_{i,s,t} \leqslant \sum_{j \in T \setminus \{i\}} (x_{i,j,s} + x_{j,i,s}) \qquad \forall i \in T, s, t \in S : s < t \leqslant \tau + s \tag{10}$$

$$y_{i,s,t} \leqslant \sum_{j \in T \setminus \{i\}} (x_{i,j,t} + x_{j,i,t}) \qquad \forall i \in T, s, t \in S : s < t \leqslant \tau + s \tag{11}$$

$$|t - u|(y_{i,t,s} + y_{j,u,s} + x_{i,j,s} - 2) \leqslant d_{i,j} \qquad \forall i, j \in T : i \neq j, s, t, u \in S : s - \tau \leqslant t, u < s \tag{12}$$

$$(\tau - (s - t - 1))\Big(x_{i,j,s} + y_{j,t,s} - 1 - \sum_{\substack{u \in S: \\ s-\tau \leqslant u < s}} \sum_{\substack{k \in T: \\ k \neq i}} (x_{i,k,u} + x_{k,i,u}) \Big) \leqslant d_{i,j} \qquad \forall i, j \in T : i \neq j, \forall s, t \in S : s - \tau \leqslant t < s \tag{13}$$

$$(\tau-(s-t-1))\Big(x_{i,j,s}+y_{i,t,s}-1-\sum_{\substack{u\in S:\\ s-\tau\leqslant u<s}}\sum_{\substack{k\in T:\\ k\neq j}}(x_{j,k,u}+x_{k,j,u})\Big)\leqslant d_{i,j} \quad \forall i,j\in T: i\neq j, \forall s,t\in S: s-\tau\leqslant t<s \quad (14)$$

$$x_{i,j,s}=0 \quad \forall i,j\in T: i\neq j, s\notin H_i\cap A_j \quad (15)$$

$$x_{i,j,s}\in\{0,1\}, d_{i,j}\geqslant 0 \quad \forall i,j\in T: i\neq j, s\in H_i\cap A_j \quad (16)$$

$$0\leqslant y_{i,s,t}\leqslant 1 \quad \forall i\in T, s,t\in S: s<t\leqslant \tau+s \quad (17)$$

The objective function (1) minimizes the sum of truncated rest differences. The first set of constraints ensures that each team plays the required number of home games against every other team, while respecting the player and venue availability constraints *(C1)* and *(C2)*. The next set of constraints enforces that a team plays at most one game per time slot. Constraints (4) and (5) recursively model the number of games a team played up to and including time slot $s \in S$, and constraints (6) limit the maximal difference in games played *(C3)*. The next two sets of constraints respectively model *(C4)* and *(C5)*. Constraints (9) to (11) regulate the value of the $y_{i,s,t}$ variables by considering the number of time slots between two consecutive games of the same team. Constraints (12) to (14) respectively model the difference in rest time when neither of the teams is fully rested, i is fully rested but j is not, and j is fully rested but i is not. Constraints (15) reduce the number of variables in the system; when implementing this formulation, these variables need not be created. Constraints (16) are the binary constraints on the x-variables and the non-negativity constraints for the d variables. Note that the integrality of $q_{i,s}$ follows from (4), (5), and (16), and that the integrality of $d_{i,j}$ and $y_{i,s,t}$ follows from the objective function and constraints (9), (10), (11), (16), and (17).

3 Pattern-based decomposition methods

A time-relaxed sports timetable can be seen as a combination of a game-off-day pattern set, a home-away pattern set, and an opponent schedule. The game-off-day pattern (GOP) of team i is a function $g_i : S \to \{G, O\}$ such that $g_i(s) = G$ if i plays a game and $g_i(s) = O$ if i has an off day (also called bye) on time slot s (see also Bao (2009)). The home-away pattern (HAP) of team i is a function $h_i : S \to \{H, A, O\}$ such that $h_i(s) = H$ if i plays a home game, $h_i(s) = A$ if i plays an away game, and $h_i(s) = O$ if i has an off day on time slot s. An assignment of one GOP to each team is known as a GOP set, and an assignment of one HAP to each team as an HAP set. The opponent schedule determines which opponent each team faces for each of the time slots. Note that the GOPs are fully defined once the HAP set or the assignment of opponents is known. Furthermore, we observe that every two patterns in an HAP set must be different, since otherwise two teams can never play against each other, while this need not be the case in a GOP set. Clearly, the assignment of opponents must be compatible with the GOP and HAP set before they can merge into a timetable: for each pair of opponents, the corresponding home-away patterns need to give one team the home advantage, and designate an away game for the other team. Moreover, if a team does not play against any opponent, it must have an off day in its game-off day pattern. We call a GOP and HAP set feasible if there exists a compatible timetable (see also Van Bulck and Goossens (2020b)).

Perhaps the most popular decomposition method in sports timetabling is the so-called first-break-then-schedule approach that breaks down timetabling by first enumerating all possible HAPs, then constructing an HAP set and checking whether it is feasible, and finally determining the opponent schedule (see Nemhauser and Trick (1998)). Despite the popularity of first-break-then-schedule, there are two issues that make it less straightforward to construct time-relaxed timetables using this approach: the combinatorial explosion of possible HAPs, and the feasibility of HAP sets.

Since in a double round-robin each team plays $(n-1)$ home games and $(n-1)$ away games, the total number of HAPs is given by $\binom{|S|}{n-1}\binom{|S|-(n-1)}{n-1}$. For 8 teams there are therefore 3.432 time-constrained HAPs, while there are 923.780 HAPs with one off day and 181.416.306.202.560 HAPs when the number of time slots is twice more than minimally needed. Hence, where the total number of time-constrained patterns is already substantial, the total number of time-relaxed HAPs quickly becomes intractable.

The efficiency of the first-break-then-schedule method heavily depends on its ability to avoid infeasible HAP sets early on (e.g., Miyashiro et al. (2003)). While it is conjectured that feasibility of a 1RR time-constrained HAP set can be verified in polynomial time (see Briskorn (2008)), it is known that verifying feasibility of a 1RR time-relaxed HAP set is $\mathcal{NP}$-complete (see Van Bulck and Goossens (2020b)).

Motivated by the following three observations, an attractive alternative to the first-break-then-schedule method seems to construct the game-off-day patterns first.

Observation 1 *Once the GOP of a team is known, it is known whether its compatible timetables respect Constraints (C1).*

Observation 2 *Once the GOP of a team is known, it is known whether its compatible timetables respect Constraints (C4) and (C5).*

Observation 3 *Once the GOP set is known, it is known whether its compatible timetables respect Constraint (C3).*

We refer to the method of first determining the GOP set and then the timetable as the *first-off-day-then-schedule* method. When constructing time-relaxed timetables, first-off-day-then-schedule seems to have several advantages over first-break-then-schedule.

First, since each team plays $2(n-1)$ games in a double round-robin tournament, there are 'only' $\binom{|S|}{2(n-1)}$ GOPs. Unfortunately, this is still a considerably large number making it impractical to enumerate all GOPs when there are many more time slots than games per team. However, the construction

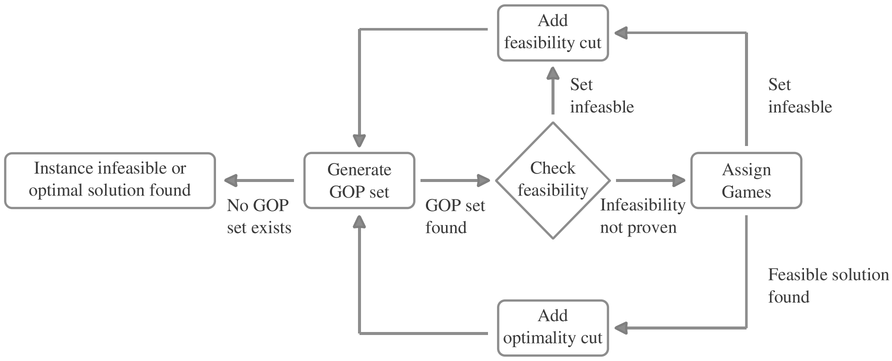

Fig. 1 General structure of the first-off-day-then-schedule algorithm. The generation of GOP sets is discussed in Section 4.1, the feasibility of GOP sets in Section 4.2, and the assignment of games in Section 4.3.

of a GOP set may be simple enough so that there is no need to explicitly enumerate all patterns first.

Second, GOP sets provide more flexibility to schedule games later on since they do not impose restrictions on the home advantage of games. Nevertheless, determining whether a given GOP set is feasible remains an $\mathcal{NP}$-complete problem, unless there is at most one off day per team (see Van Bulck and Goossens (2020b)).

While the conceptual idea of first-off-day-then-schedule was to some extent already outlined by Bao (2009), Bao does not provide any computational experiments with regard to the performance of the first-off-day-then-schedule method. Moreover, Bao does not provide any information on how to backtrack when one of the subproblems turns out to be infeasible.

4 Implementing first-off-day-then-schedule

In order to implement a first-off-day-then-schedule approach for the problem outlined in Section 2, we implement a GOP set generating decomposition method (see Figure 1). Our approach decomposes the problem into the following three components: generate a GOP set (Section 4.1), check feasibility of the GOP set (Section 4.2), and find a timetable compatible with the GOP set (Section 4.3). If one of the subproblems turns out to be infeasible, we implement backtracking. If the master problem is infeasible, then the last solution found is optimal, or in case no solution was found the problem instance is infeasible.

4.1 Generating GOP sets

In order to generate GOP sets with constraint programming (CP), we formulate the GOP set generation model (18)-(27). Our main decision variable is

$g_{i,s}$ which is 1 if team $i \in T$ has a 'G' on time slot $s \in S$, and 0 otherwise. The global constraint sequence$(nbMin, nbMax, width, vars, values, card)$ takes six arguments: $values$ and $card$ are arrays of integers and must have the same index set I, $vars$ is an array of decision variables, and the remaining arguments are integers. For each $i \in I$ the constraint requires that $card[i]$ elements of $vars$ take value $value[i]$, and that any subsequence of size $width$ must contain at least $nbMin$ and at most $nbMax$ values from $values$ (see CP Optimizer (2014)).

$$\text{sequence}\big(0, 2, \rho, (g_{i,1}, \dots, g_{i,|S|}), (1), (2n-2)\big)\big) \qquad \forall i \in T \quad (18)$$

$$\text{sequence}\big(0, \sigma, \sigma+1, (g_{i,1}, \dots, g_{i,|S|}), (0), (|S|-2n+2)\big)\big) \qquad \forall i \in T \quad (19)$$

$$\sum_{s \in H_i} g_{i,s} \geqslant n-1 \qquad \forall i \in T \quad (20)$$

$$\sum_{s \in H_i \cap A_j} (g_{i,s} = 1 \wedge g_{j,s} = 1) \geqslant 1 \qquad \forall i, j \in T : i \neq j \quad (21)$$

$$\sum_{\substack{s \in (H_i \cap A_j) \\ \cup (H_j \cap A_i)}} (g_{i,s} = 1 \wedge g_{j,s} = 1) \geqslant 2 \qquad \forall i, j \in T : i < j \quad (22)$$

$$\Big|\sum_{p=1}^{s} g_{i,p} - \sum_{p=1}^{s} g_{j,p}\Big| \leqslant \text{GPDI} \qquad \forall i, j \in T : i < j, \forall s \in S \quad (23)$$

$$\Big(\sum_{i \in T} g_{i,s} \bmod 2\Big) = 0 \qquad \forall s \in S \quad (24)$$

$$\sum_{i \in T : s \in H_i} g_{i,s} \geqslant \sum_{i \in T : s \in A_i \setminus H_i} g_{i,s} \qquad \forall s \in S \quad (25)$$

$$g_{i,s} = 0 \qquad \forall i \in T, s \in S \setminus A_i \quad (26)$$

$$g_{i,s} \in \{0,1\} \qquad \forall i \in T, s \in S \quad (27)$$

Constraints (18) model that each GOP contains $2(n-1)$ 'G's, and that no GOP contains more than 2 games in any sequence of ρ time slots *(C4)*. Constraints (19) model that each GOP contains $|S| - 2(n-1)$ off days, and that no GOP contains more than σ consecutive off days *(C5)*. Since each team has to play $(n-1)$ home games, the next set of constraints requires that each GOP contains $(n-1)$ 'G's during time slots on which the venue of the team is available. Constraints (21) enforce that each game (i,j) can be scheduled, and Constraints (22) enforce that game (i,j) and (j,i) can be scheduled simultaneously (at least when ignoring all other games). Constraints (23) limit the maximal difference in games played *(C3)*. The next set of constraints enforces that the sum of 'G's is even on each time slot, a necessary condition since each game involves two teams. Its simplicity notwithstanding, this necessary condition is sufficient for the feasibility of a 1RR GOP set if each team has at most one off day (see Van Bulck and Goossens (2020b)). Since each game involves one home team and one away team, Constraints (25) enforce that the total number of teams that can play home and have a 'G' on time slot $s \in S$ must be larger than or equal to the number of teams that have a 'G' and can only play away on s. Finally, Constraints (26) enforce that a team has an off day when its players are unavailable, and Constraints (27) are the binary constraints.

In order to increase the probability that there exists a compatible timetable for the generated GOP set, we may replace the right-hand side of Constraints (21) by $1+\epsilon$ where ϵ is a positive integer parameter which is reduced by one if no feasible solution for formulation (18)-(27) is found. Note that Constraints (22) become redundant if $\epsilon \geq 1$.

4.2 Feasibility checks

A GOP set is feasible if all games can be assigned to time slots on which the opposing teams have a ‘G’ in their pattern and the venue of the home team is available. In case a GOP set turns out to be infeasible, we add a constraint that prevents from finding the infeasible GOP set again. We thereby try to reduce the total number of solutions that need to be enumerated in the master problem by cutting-off as many infeasible or sub-optimal solutions as possible (see also Rasmussen and Trick (2007)).

4.2.1 Game possibilities

Denote with $c_G(T', s)$ the number of ‘G’s in the GOPs of a subset of teams $T' \subseteq T$, $|T'| = m$, on time slot $s \in S$. The number of games between teams in T' on s is at most $\lfloor \frac{c_G(T',s)}{2} \rfloor$. Hence, Condition 1 is a necessary condition that requires to check $\mathcal{O}(2^n)$ constraints.

Condition 1 (Bao (2009)) $\sum_{s \in S} \left\lfloor \frac{c_G(T',s)}{2} \right\rfloor \geqslant m(m-1)$ *for each* $T' \subseteq T$.

Instead of explicitly checking Condition 1 for each subset, we formulate an IP model to find a minimal subset of teams for which Condition 1 is violated, or which proves that no such subset exists. The formulation of this IP is based on Rasmussen and Trick (2007), with the main difference that it checks the feasibility of a GOP set instead of an HAP set and that it requires to be solved only once instead of once for each cardinality of T'. Parameter $\mathrm{UB}_{\mathrm{GP}}$ gives an upper bound on the cardinality of the subsets to be checked, and parameters $g'_{i,s}$ define the GOP set found by model (18)-(27). The main purpose of parameter $\mathrm{UB}_{\mathrm{GP}}$ is to control for the expected computation time to solve the IP formulation. Variable z_c, $2 \leqslant c \leqslant \mathrm{UB}_{\mathrm{GP}}$, determines whether the cardinality of the subset is c ($z_c = 1$) or not ($z_c = 0$), α_i determines whether team i is in the subset ($\alpha_i = 1$) or not ($\alpha_i = 0$), and variable β_s, $s \in S$, calculates an upper bound on the total number of games the teams in the subset can play.

$$\textbf{minimize} \quad \sum_{i \in T} \alpha_i \tag{28}$$

$$\sum_{i \in T} \alpha_i = \sum_{c=2}^{\mathrm{UB_{GP}}} c z_c \tag{29}$$

$$\sum_{c=2}^{\mathrm{UB_{GP}}} z_c = 1 \tag{30}$$

$$\beta_s \geqslant \big(\sum_{i \in T} g'_{i,s} \alpha_i - 1 \big)/2 \qquad \forall s \in S \tag{31}$$

$$\sum_{s \in S} \beta_s \leqslant c(c-1) - 1 + \mathrm{UB_{GP}}(\mathrm{UB_{GP}} - 1)(1 - z_c) \qquad \forall c \in \mathbb{N} : 2 \leqslant c \leqslant \mathrm{UB_{GP}} \tag{32}$$

$$\alpha_i, z_c \in \{0,1\}, \beta_s \in \mathbb{N} \qquad \forall i \in T, s \in S, c \in \mathbb{N} : 2 \leqslant c \leqslant \mathrm{UB_{GP}} \tag{33}$$

The objective function minimizes the total number of teams chosen such that infeasibility of the GOP set can be traced back to as few patterns as possible, and Constraints (29)-(30) model the value of z_c. Constraints (31) calculate the upper bound on the total number of games between teams in the subset on time slot s, and Constraints (32) ensure that Condition 1 is violated. Finally, Constraints (33) are the binary and integrality constraints.

If a violating set of teams T' defined by the α_i's is found, the following constraint is added to the GOP set generation model.

$$\mathrm{forbiddenAssignment}\big((g_{i,1}, \dots, g_{i,|S|} \, \forall i \in T'), (g'_{i,1}, \dots, g'_{i,|S|} \, \forall i \in T')\big) \tag{34}$$

The global constraint forbiddenAssignment($vars, values$) takes two arguments: an array of decision variables $vars$ and an ordered set $values$ that both have index set I. The constraint enforces there is at least one $i \in I$ such that $vars[i]$ is not equal to $values[i]$. Intuitively this constraint forbids any GOP set in which the teams in T' play according to the GOPs currently assigned. Clearly, the smaller is $|T'|$, the stronger is the reduction in the search space.

4.2.2 Isolated slots

Define with $S_{T'} \subseteq S$ the subset of time slots on which at least two teams in $T' \subseteq T$, $|T'| = m$, have a game and all teams not in T' have an off day, i.e. $\sum_{i \in T'} g'_{i,s} \geqslant 2$ and $\sum_{i \notin T'} g'_{i,s} = 0$ for all $s \in S_{T'}$. We refer to the subset of time slots $S_{T'}$ as isolated slots for the subset of teams T'. Note that it follows from the definition that S_T contains all time slots on which at least two teams have a 'G' in their pattern.

Condition 2 *For each subset of teams $T' \subseteq T$, the sum of G's during isolated slots is smaller than twice the total number of mutual games in T', i.e. $\sum_{i \in T'} \sum_{s \in S_{T'}} g'_{i,s} \leqslant 2m(m-1)$.*

Condition 2 is a necessary condition since teams in T' can only play against other teams in T' during isolated slots, and the total number of games between teams in T' is limited by $m(m-1)$.

Instead of explicitly checking each subset, we formulate an IP model to find a minimal subset of teams for which Condition 2 is violated, or which proves that no such subset exists. In this formulation, parameter $\mathrm{UB_{IS}}$ denotes the maximal subset to be checked. Furthermore, let γ_s, $s \in S_T$, denote the number of 'G's if s is an isolated slot for the subset of teams defined by α_i.

$$\textbf{minimize} \quad \sum_{i \in T} \alpha_i \tag{35}$$

$$\sum_{i \in T} \alpha_i = \sum_{c=2}^{\mathrm{UB_{IS}}} c z_c \tag{36}$$

$$\sum_{c=2}^{\mathrm{UB_{IS}}} z_c = 1 \tag{37}$$

$$\gamma_s \leqslant \alpha_i \sum_{j \in T} g'_{j,s} \qquad \forall s \in S_T, i \in T : g'_{i,s} = 1 \tag{38}$$

$$\sum_{s \in S_T} \gamma_s \geqslant 2c(c-1) + 1 - \big(\mathrm{UB_{IS}}(\mathrm{UB_{IS}} - 1) + 1\big)(1 - z_c) \qquad \forall c \in \mathbb{N} : 2 \leqslant c \leqslant \mathrm{UB_{IS}} \tag{39}$$

$$\alpha_i, z_c \in \{0, 1\}, \gamma_s \geq 0 \qquad \forall i \in T, s \in S_T, c \in \mathbb{N} : 2 \leqslant c \leqslant \mathrm{UB_{IS}} \tag{40}$$

The objective function minimizes the total number of teams in the subset, while Constraints (36) and (37) model the z_c variables. If all teams that have a 'G' on time slot $s \in S_T$ are in the subset, s is an isolated slot and Constraints (38) counts the total number of 'G's. Constraints (39) ensure that Condition 2 is violated, and Constraints (40) are the binary and non-negativity constraints.

If a violating set of teams T' defined by the α_i's is found, the following constraint is added to model (18)-(27).

$$\text{forbiddenAssignment}\big((g_{1,s}, \ldots, g_{|T|,s}\, \forall s \in S'), (g'_{1,s}, \ldots, g'_{|T|,s}\, \forall s \in S')\big) \tag{41}$$

Constraint (41) forbids any GOP set in which the teams have a 'G' according to the current columns in S'.

4.3 Assign games

Given a GOP set, a third and final step is to construct a compatible timetable. This section first provides two more feasibility checks based on the construction of a compatible timetable, and then shows how to construct a compatible timetable that minimizes the sum of truncated rest differences.

Consider first the following condition for the feasibility of a GOP set.

Condition 3 *For each subset of time slots $S' \subseteq S$, an assignment of games to time slots in S' exists such that for each $s \in S'$ team $i \in T$ plays exactly one* *game (i,j) or (j,i) if $g'_{i,s}=1$ and $s \in H_i$, exactly one game (i,j) if $g'_{i,s}=1$ and $s \in A_i \setminus H_i$, and no game if $g'_{i,s}=0$.*

By the definition of a feasible GOP set, Condition 3 is a necessary condition for all $S' \subseteq S$ and a sufficient condition if $S' = S$. In essence, Condition 3 checks feasibility for a subset of columns in the GOP set. We use the linear relaxation of formulation (42)-(45) to check Condition 4 for each S' with cardinality $\mathrm{UB}_{\mathrm{AGC}}$ or lower, and add a constraint of type (41) to the GOP set generation model if a violating subset of time slots S' is found. The main motivation for the use of the linear relaxation is the expected decrease in computation time, while (hopefully) still detecting violations of Condition 3 reasonably well.

$$\sum_{s \in S' \cap H_i} x_{i,j,s} \leq 1 \qquad \forall i,j \in T : i \neq j \quad (42)$$

$$\sum_{j \in T \setminus \{i\}} (x_{i,j,s} + x_{j,i,s}) = g'_{i,s} \qquad \forall i \in T, \forall s \in S' \quad (43)$$

$$x_{i,j,s} = 0 \qquad \forall i \in T, \forall s \in S' : s \notin H_i \vee g'_{i,s} = 0 \vee g'_{j,s} = 0 \quad (44)$$

$$x_{i,j,s} \in \{0,1\} \qquad \forall i,j \in T : i \neq j, \forall s \in S' \quad (45)$$

The first set of constraints restricts each game (i,j) to be scheduled at most once, and the second set of constraints enforces the GOP set for the given subset of time slots. Finally, Constraints (44) reduce the number of variables in the system, and Constraints (45) are the binary constraints.

Instead of checking the columns of a GOP set, we may also check feasibility for a subset of rows in the GOP set (see also Rasmussen and Trick (2007)).

Condition 4 *For each subset of teams $T' \subseteq T$, an assignment of the mutual games between teams in T' to time slots in S exists in which the opposing teams have a 'G' in their pattern and the venue of the home team is available.*

By the definition of a feasible GOP set, Condition 4 is a necessary condition for all $T' \subseteq T$ and a sufficient condition if $T' = T$. We use the linear relaxation of formulation (46)-(50) to check Condition 4 for each subset of teams with cardinality $\mathrm{UB}_{\mathrm{AGR}}$ or lower, and add a constraint of type (34) to the GOP set generation model if a violating subset of teams T' is found. Observing that the rest time of teams is known once the GOP set is known, parameter $r_{i,s}$ gives the truncated rest time of team $i \in T'$ in time slot $s \in S$. Parameter obj^*, initially equal to infinity, gives the objective value of the current best found solution.

$$\sum_{s \in H_i} x_{i,j,s} = 1 \qquad \forall i,j \in T' : i \neq j \quad (46)$$

$$\sum_{j \in T' \setminus \{i\}} (x_{i,j,s} + x_{j,i,s}) \leqslant 1 \qquad \forall i \in T', \forall\, s \in S : g'_{i,s} = 1 \quad (47)$$

$$\sum_{\substack{i,j \in T': \\ i \neq j}} \sum_{s \in S} |r_{i,s} - r_{j,s}| x_{i,j,s} \leqslant \text{obj}^* - 1 \quad (48)$$

$$x_{i,j,s} = 0 \qquad \forall i \in T', \forall s \in S : s \notin H_i \vee g'_{i,s} = 0 \vee g'_{j,s} = 0 \quad (49)$$

$$x_{i,j,s} \in \{0,1\} \qquad \forall i,j \in T' : i \neq j, \forall\, s \in S \quad (50)$$

The first set of constraints enforces that each team in T' plays exactly once at home against every other team in T', and the second set of constraints enforces that each team in T' plays at most one game per time slot. Constraint (48) states that the sum of truncated rest differences must be better than the current best found solution. Finally, Constraints (49) reduce the number of variables in the system, and Constraints (50) are the integrality constraints.

In case no subset of time slots or teams is found that respectively violates Condition 3 or Condition 4, we enable all integrality constraints and we solve formulation (46)-(50) for $T' = T$. In addition, we minimize the following objective.

$$\textbf{minimize} \quad \sum_{\substack{i,j \in T': \\ i \neq j}} \sum_{s \in S} |r_{i,s} - r_{j,s}| x_{i,j,s} \quad (51)$$

If a solution exists, we have found a strictly better solution and we update obj* . Moreover, regardless of the feasibility, we add a constraint of type (34) to the GOP set generation model to prevent that the same solution is found again.

5 Computational Results

This section experimentally evaluates IP formulation (1)-(17) and the first-off-day-then-schedule approach respectively proposed in Section 2 and Section 4.

Our benchmark of problem instances consists of 9 artificial double round-robin problem instances. A problem instance in this set is of type (n, o, h, a) if it contains n teams and $2(n-1)+o$ time slots (i.e., each team has o off days), and for each team $i \in T$ it holds that $|H_i| = h$ and $|S \setminus A_i| = a$ (see also Van Bulck and Goossens (2020a)). We consider n in the set $\{8, 12, 16\}$, o in $\{(n-1), 2(n-1), 3(n-1)\}$, set $h = o/2$ and $a = o/4$, and assume that a team is fully rested after five time slots (i.e. $\tau = 5$). Furthermore, we require that the games-played difference index is not larger than 2 (i.e. GPDI $= 2$), and that a team has at most six consecutive off days (i.e. $\sigma = 6$). Finally, if the total number of time slots is at least twice the minimal number needed we

				Algorithms	
Teams n	o	h	a	IP	FOTS
8	21	3	1	8	16
	28	7	3	8	29
	35	10	5	38	42
12	33	5	2	71	45
	44	11	5	133	98
	55	16	8	205	143
16	45	7	3	185	76
	60	15	7	/	155
	75	22	11	320	267

Table 1 Results for the artificial problem instances. The first four columns respectively refer to the number of teams (n), the number of off days per team (o), the average venue availability (h), and the average player unavailability (a). Algorithm 'IP' refers to solving integer programming formulation (1)-(17), and algorithm 'FOTS' refers to the first-off-day-then-schedule approach outlined in Section 4. Finally, '/' means that no solution was found within the given computation time.

require that no team plays more than 2 games within 3 consecutive time slots (i.e. $\rho = 3$) and set $\rho = 2$ otherwise.

In order to define the parameters of the first-off-day-then-schedule algorithm, we require that each game can be scheduled during at least four time slots (i.e. $\epsilon = 3$), check feasibility of Condition 1 and Condition 2 for up to 6 teams (i.e. $\mathrm{UB}_{\mathrm{GP}} = 6$ and $\mathrm{UB}_{\mathrm{IS}} = 6$), and check feasibility of Condition 3 and Condition 4 for up to 4 time slots and 4 teams (i.e. $\mathrm{UB}_{\mathrm{AGC}} = 4$ and $\mathrm{UB}_{\mathrm{AGR}} = 4$).

All IP formulations are solved with ILOG CPLEX version 12.10, and the CP formulation is solved with ILOG CPLEX CP OPTIMIZER 12.10. The IP formulation (1)-(17) was granted 3600 seconds of computation time, whereas the first-off-day-then-schedule approach was granted only 600 seconds of computation time. All models were run on a CentOS 7.4 GNU/Linux based system with an Intel E5-2680 processor, running at 2.5 GHz and provided with 16 GB of RAM and 8 cores.

Table 1 presents preliminary results for the best found solution by each algorithm. The first four columns provide the total number of teams (n), the number of off days per team (o), the average team availability (h), and the average player unavailability (a). The next column gives the best found solution within the given computation time using IP formulation (1)-(17); none of the instances was solved to optimality and the best lower bound found was equal to 0 for all problem instances. The final column shows the best found solution by our first-break-then-schedule algorithm. Despite being given 6 times less computation time, the first-off-day-then-schedule finds solutions that are only slightly worse when there are 8 teams in the competition, and finds even better solutions for all problem instances that have more than 8 teams.

6 Conclusion

Decomposition methods are common in sports timetabling and break down the timetabling process into different subproblems. Perhaps the most popular decomposition method is first-break-then-schedule where the first subproblem is to determine the home-away pattern (HAP) set which defines when teams must play at home or play away. Observing that existing decomposition methods focus exclusively on time-constrained timetables where the number of time slots is the minimally needed, this paper investigates how to generalize first-break-then-schedule so as to construct time-relaxed timetables where there are more time slots than games per team. In particular, this paper proposes a first-off-day-then-schedule method that first determines the game-off-day pattern (GOP) set defining when teams play (home or away) or have an off day, after which it constructs a compatible timetable.

The main advantage of our method is that the construction of GOP sets turns out to be simple enough so that there is no need to explicitly enumerate all patterns first. This avoids a combinatorial explosion by only implicitly enumerating all patterns, and allows to check feasibility of GOP sets both on the level of rows (representing patterns of teams) and columns (representing the teams that play on a particular time slot). Nevertheless, our approach still generates many infeasible GOP sets and would therefore profit from further research on the level of GOP set feasibility.

We use our approach to generate a number of time-relaxed double round-robin timetables where availability constraints and fairness issues play a prominent role. Indeed, while the structure of a round-robin tournament already adds a substantial level of fairness to any timetable designed for it, there are many other fairness issues that need to be considered. In this paper, we minimize the sum of rest differences while controlling for the rest period between teams' consecutive games and the maximal difference in games played.

References

Atan T, Çavdaroğlu B (2018) Minimization of rest mismatches in round robin tournaments. Comput Oper Res 99:78 – 89

Bao R (2009) Time relaxed round robin tournament and the NBA scheduling problem. PhD thesis, Cleveland State University

Bengtsson H, Ekstrand J, Hägglund M (2013) Muscle injury rates in professional football increase with fixture congestion: an 11-year follow-up of the UEFA Champions League injury study. Br J Sports Med 47:743–747

Briskorn D (2008) Feasibility of home-away-pattern sets for round robin tournaments. Oper Res Lett 36:283 – 284

Çavdaroğlu B, Atan T (2020) Determining matchdays in sports league schedules to minimize rest differences. Oper Res Lett 48:209 – 216

CP Optimizer II (2014) 12.6, IBM ILOG CPLEX Optimization Studio CP Optimizer user's manual, 2014

Dupont G, Nedelec M, McCall A, McCormack D, Berthoin S, Wisløff U (2010) Effect of 2 soccer matches in a week on physical performance and injury rate. Am J Sport Med 38:1752–1758

Goossens D, Spieksma F (2009) Scheduling the Belgian soccer league. Interfaces 39:109–118

Miyashiro R, Iwasaki H, Matsui T (2003) Characterizing feasible pattern sets with a minimum number of breaks. In: Burke E, De Causmaecker P (eds) Practice and Theory of Automated Timetabling IV, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 78–99

Nemhauser GL, Trick MA (1998) Scheduling a major college basketball conference. Oper Res 46:1–8

Rasmussen RV (2008) Scheduling a triple round robin tournament for the best Danish soccer league. Eur J Oper Res 185:795 – 810

Rasmussen RV, Trick MA (2007) A benders approach for the constrained minimum break problem. Eur J Oper Res 177:198 – 213

Schönberger J, Mattfeld DC, Kopfer H (2004) Memetic algorithm timetabling for non-commercial sport leagues. Eur J Oper Res 153:102–116

Scoppa V (2015) Fatigue and team performance in soccer: Evidence from the FIFA World Cup and the UEFA European Championship. J Sport Econ 16:482–507

Suksompong W (2016) Scheduling asynchronous round-robin tournaments. Oper Res Lett 44:96–100

Trick MA (2001) A schedule-then-break approach to sports timetabling. In: Burke E, Erben W (eds) Practice and Theory of Automated Timetabling III, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 242–253

Van Bulck D, Goossens D (2020a) Handling fairness issues in time-relaxed tournaments with availability constraints. Comput Oper Res 115:104856

Van Bulck D, Goossens D (2020b) On the complexity of pattern feasibility problems in time-relaxed sports timetabling. Oper Res Lett 48:452 – 459

Van Bulck D, Goossens DR, Spieksma FCR (2019) Scheduling a non-professional indoor football league: a tabu search based approach. Ann Oper Res 275:715–730

Van Bulck D, Goossens D, Schönberger J, Guajardo M (2020) RobinX: A three-field classification and unified data format for round-robin sports timetabling. Eur J Oper Res 280:568 – 580