RobinX: a three-field classification and unified data format for round-robin sports timetabling

David Van Bulck[a,*], Dries Goossens[a], Jörn Schönberger[b], Mario Guajardo[c],

[a]*Ghent University, Faculty of Economics and Business Information*
[b]*Technische Universität Dresden, Institute of Transport and Economics*
[c]*NNH Norwegian School of Economics, Department of Business and Management Science*

Abstract

Sports timetabling problems are combinatorial optimization problems which consist of creating a timetable that defines against whom, when, and where teams play games. In the literature, sports timetabling problems have been reported featuring a wide variety of constraints and objectives. This variety makes it challenging to identify the relevant set of papers for a given sports timetabling problem. Moreover, the lack of a generally accepted data format makes that problem instances and their solutions are rarely shared. Consequently, it is hard to assess algorithmic performance since solution methods are often tested on just one or two specific instances. To mitigate these issues, this paper presents RobinX, a three-field notation to describe a sports timetabling problem by means of the tournament format, the constraints in use, and the objective. We use this notation to classify sports timetabling problems presented in the operations research literature during the last five decades. Moreover, RobinX contains XML-based file templates to store problem instances and their solutions and presents an online platform that offers three useful tools. First, a query tool assists users to select the relevant set of papers for a given timetabling problem. Second, the online platform provides access to an XML data repository that contains real-life problem instances from different countries and sports. Finally, the website enables users to interact with a free and open-source C++-library to read and write XML files and to validate and evaluate encoded instances and solutions.

Keywords: OR in sports, Validation of OR Computations, Sports scheduling, RobinX, XML

1. Introduction

Creating timetables for sports competitions has been a topic of research since the 1970s (e.g., Ball and Webster (1977)). Ever since, academic papers about sports timetabling have increased considerably in numbers and sports timetabling has become a specialized field with its own research conferences (Kendall et al., 2010). Figure 1 illustrates the increase in operations research related sports timetabling contributions over the last five decades. The peak in contributions around 2000 is partially explained by the introduction of the traveling tournament problem (Easton et al. (2001)), which minimizes the total team travel in a timetable. For this problem, substantial algorithmic progress has been reported after Easton et al. (2001) made a set of artificial benchmark instances publicly available. Many of the other contributions read as a case study that constructs the timetable of a single real-life competition. This is a complex matter due to conflicting interests of many stakeholders: apart from some basic constraints, each competition has its own requirements. For example, minimizing the occurrence of two consecutive home games or two consecutive away games for the same team is often key in professional competitions (e.g., Bartsch et al. (2006); Goossens and Spieksma (2009)). In contrast, respecting player availability is far more important in non-professional competitions (e.g., Schönberger et al. (2004); Van Bulck et al. (2019)). This wide variety of objectives and requirements makes it challenging to identify the relevant set of papers for a given timetabling problem. Moreover, the lack of a generally accepted data format makes that (real-life) problem instances and their solutions are rarely shared. Consequently, contributions in the literature are often tested on just one or two specific problem instances of a particular competition. This forms one of the main obstacles in current algorithmic progress for solution methods since few general insights have been gained from previous studies.

In a round-robin timetable, often imprecisely called a schedule (for an argumentation see, e.g., Schreuder (1992); Schönberger et al. (2004)), every team plays against every other team a fixed number of times. Although a few contributions have been made to organize various constraints that occur in (single-league) round-robin timetabling, they did not result in a generally applicable notation or file format for problem instances and solutions. Bartsch et al. (2006) describe several sports competitions by means of organizational constraints that enforce the various competition rules imposed by the league organizers, attractiveness constraints that make the competition thrilling over the entire season, and fairness constraints that guarantee that the timetable does not favor any team. Ras-

[*]Corresponding author
Email address: david.vanbulck@ugent.be (David Van Bulck)

This is the peer-reviewed author-version of https://doi.org/10.1016/j.ejor.2019.07.023*, published in European Journal of Operational Research.*

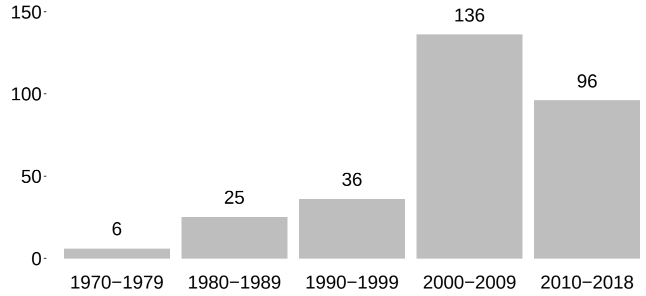

Figure 1: Sports timetabling contributions from Knust (2018) categorized according to the year of publication.

mussen and Trick (2008) list eight common constraint classes but do not describe what particular constraints within each class look like. Nurmi et al. (2010) elaborate further on previous contributions and list more than thirty popular round-robin constraints. Moreover, they are the first to set up a (plain text-only) file format to store instances, and propose a set of artificial and real-world instances together with the best solutions found so far. This format, however, has limited utility with respect to the ease of data manipulation and is not extensible towards several real-world problems such as the multi-league sports timetabling problem. Kendall et al. (2010) are the only authors to distinguish between different objectives.

The main contributions and outline of this paper are as follows. Section 2 formally defines the structure of multi-league sports timetabling problems, and introduces the most common sports timetabling terminology. As a first contribution, Section 3 then proposes a three-field notation to describe (*i*) the tournament format (*ii*) the constraints in use and (*iii*) the objective of a sports timetabling problem instance. The notation is able to classify a wide variety of round-robin tournaments and other competition formats. This unified notation should help researchers to recognize common problem features among different problem instances. As a second contribution, Section 4 provides a problem-driven classification of sports timetabling applications presented in the operations research literature. Based on this classification, an interactive query tool is proposed to assist researchers to identify all relevant papers for a given timetabling problem. As a third contribution, Section 5 encodes the three-field notation using XML-based file templates to store problem instances, solutions, and objective bounds. While still being human readable, the advantage of XML over plain text-only file formats lies in the structured way of data storage making it an extendable language that can easily be adjusted over time. Another advantage of the XML files is the ability to retrieve problem specific properties that may have been lost in the three-field notation which operates on a higher abstraction level. To encourage further research, we provide a `C++`-library to read and write XML files and to validate and evaluate encoded instances and solutions. In addition, we employ this library to encode over 40 different problem instance types originating from over 15 different countries and eight different sports. All collected data and several related tools are available on the website devoted to this project (Van Bulck et al., 2018b). Conclusions follow in Section 6.

2. Terminology

The input of a sports timetabling problem consists of a set of time slots P, a set of teams U, and a multiset of games G. Time slots can represent periods in time like half days or days in the season, however, a team can never play more than one game per time slot. If a team does not play in a time slot, it has a *bye* in that time slot. To represent relations between time slots, a time group $S \subseteq P$ combines multiple time slots, e.g., all midweek time slots, into a single set. Time groups can also represent so-called rounds that consist of a set of one or more time slots during which a team can play at most one game. Although there is a temporal relationship between time slots, s_i is earlier in time than s_{i+1}, there is no such relation between time groups. The multiset of games G consists of ordered pairs (i, j) in which $i \in U$ is the home team providing the venue where the game is played, and $j \in U$ is the away team. For convenience, we denote with $g_{i,j}$ the multiplicity of the ordered pair $(i, j) \in G$, i.e., $g_{i,j}$ gives the number of home games i has to play against j. Likewise, we denote with g_i the total number of games in G involving team i. Similar to time groups, teams can be combined into a team group $T \subseteq U$ to represent an entity such as a sports club, a strength group, or a geographical group. Team groups are also useful to model multi-league sports timetabling problems in which a set of leagues L form a partition of the teams U, i.e., $\bigcup_{l \in L} T_l = U$ and $T_{l_1} \cap T_{l_2} = \emptyset$ for $l_1 \neq l_2$, such that $g_{i,j}$ is zero if i and j do not belong to the same league, and possibly non-zero otherwise. A timetable maps each game in G to a time slot $s \in P$ such that no team plays more than one game per time slot. Sometimes, we can also associate a cost $c_{i,j,s}$ with playing game $(i, j) \in G$ on time slot $s \in P$.

Tournament format. Team group T organizes a k round-robin tournament (kRR) if $g_{i,j} + g_{j,i}$ equals a constant k for each $i, j \in T$ with $i \neq j$ and the difference between $g_{i,j}$ and $g_{j,i}$ is at most one. Similarly, team group T organizes a k bipartite round-robin tournament (kBRR) if the teams in T can be partitioned into two disjoint sets V_1 and V_2 in such a way that $g_{i,j} + g_{j,i}$ equals a constant k if $i \in V_1$ and $j \in V_2$ and the difference between $g_{i,j}$ and $g_{j,i}$ is at most one, and 0 otherwise. In any other case, we say that the team group organizes a non round-robin tournament (NRR). A tournament is *compact* or *time-constrained* if the number of available time slots $|P|$ is no more than the minimum number required to play all games in that tournament. In a kRR with n teams, n even, the minimum number of time slots to play all games equals $k(n-1)$; if n is odd, the minimum number of time slots is kn. In a kBRR, this number equals k times the number of teams in the largest of two team groups, that is $k \max(|V_1|, |V_2|)$. A *time-relaxed* tournament has more time slots available than strictly needed. Table 1 gives an example of a timetable for a compact 2RR.

Symmetry structures. In a kRR with $k > 1$, the season is often split into k intervals, i.e., a series of consecutive time slots

Table 1: A compact mirrored double round-robin timetable for a single league with 6 teams.

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
(1,2)	(2,5)	(2,4)	(2,3)	(6,2)	(2,1)	(5,2)	(4,2)	(3,2)	(2,6)
(3,4)	(4,1)	(1,6)	(5,1)	(4,5)	(4,3)	(1,4)	(6,1)	(1,5)	(5,4)
(5,6)	(6,3)	(5,3)	(6,4)	(1,3)	(6,5)	(3,6)	(3,5)	(4,6)	(3,1)

of length $|P|/k$ that each contain a 1RR. We call a timetable that follows this format *phased* and consider the following additional symmetry structures (see Table 2). In the *mirrored* timetable format, the opponents in each interval are identical to the opponents of the previous interval. In the *inverted* system, intervals are played in the reversed order of the previous interval. In the *English* system, the opponents in the first time slot of an interval are the same as in the last time slot of the previous interval, and the opponents of the l-th time slot of the interval correspond with opponents of the $(l-1)$-th time slot of the previous interval. Finally, in the *French* system, the opponents in the last time slot of an interval correspond with the opponents in the first time slot of the previous interval. For all other time slots, the opponents of the l-th time slot in in an interval correspond with the opponents in the $(l+1)$-th time slot of the previous interval. An overview of symmetry structures in European top football competitions (soccer in the USA) can be found in Goossens and Spieksma (2012).

Fairness concepts. A team has a *home stand* if it plays multiple home games in a row and is on a *road trip* when it plays multiple away games in a row. If a team plays a game with the same home-away status as its previous game, no matter the number of byes in between, we say it has a *break*. As an example, team 2 in Table 1 has a home break on time slots s_3 and s_4 and has a home stand starting on s_2 and ending on s_4. When a team first plays against team i, and immediately thereafter against team j, we say that team i gives a carryover effect (coe) to team j (see e.g., Russell (1980)). If we denote with $c_{i,j}$ the number of carryover effects that team i gives to team j over the entire tournament, then the carryover effect value (coe-value) of a timetable is defined as $\sum_{i\in U}\sum_{j\in U} c_{i,j}^2$. Note that the coe-value is a cyclical concept: it also considers the carryover from a team's last game to its first game. For fairness reasons, it is sometimes requested that the coe-value of a team group is as low as possible. Indeed, if i is very strong team, one might believe that j has an advantage since j's opponent is more likely to be weakened or injured. A timetable for a kRR with n teams has a coe-value of at least $kn(n-1)$; a timetable that realizes this lower bound is called coe-balanced. A weighted variant of the carryover effect in which each $c_{i,j}^2$ is additionally multiplied with weight $w_{i,j}$ is proposed in Guedes and Ribeiro (2011).

Timetable parameters. Finally, for a given timetable, we define the following parameters. Please note that these parameters are solely used to describe how constraint violations for a given timetable should be evaluated (see Section 3.2); we do not intend to use them as decision variables in a mathematical model. First, for each team pair in U, $x_{i,j,s}$ is 1 if team i and j meet in the venue of i on time slot $s \in P$, and 0 otherwise. Similarly, $o_{i,j,k}$ is 1 if the k-th opponent of team i is team j, and 0 otherwise. The parameter $h_{i,j,k}$ is 1 if i and j meet for the k-th time in the venue of i, and 0 otherwise. Parameter y_{i,j,s_1,s_2} is 1 if team i and j meet each other during time slots $s_1 \in P$ and $s_2 \in P$, $s_1 < s_2$, without meeting in between, and 0 otherwise. Furthermore, let $h_{i,s}$ be 1 if team i plays a home game on time slot s, and 0 otherwise. Similarly, $q_{i,k}$ is 1 if team i plays its k-th game at home, and 0 otherwise. Let $b_{i,s}$ be 1 if team i has a break on time slot s, and 0 otherwise. In the weighted break minimization problem, break $b_{i,s}$ is additionally multiplied with weight $v_{i,s}$. Also, let a_{i,s_1,s_2} be 1 if team i starts a road trip on s_1 and ends the trip on s_2, and 0 otherwise. Finally, $e_{i,s}$ is equal to the distance traveled by i from the venue of its previous game to the venue of its current game if i plays a game on time slot s, and 0 otherwise. We assume that each team is located at its home venue before the season starts, and needs to return back home immediately after playing its last game. Therefore, $e_{i,s}$ also contains the cost of returning back home if team i plays its last game on time slot s. For a complete overview of sports timetabling terminology, we refer to Drexl and Knust (2007), Kendall et al. (2010), and Rasmussen and Trick (2008).

3. Three-field notation

In the late 1970s, Graham et al. (1979) introduced a now widely used three-field notation to distinguish between different machine scheduling problems, in which a set of tasks has to be sequenced and assigned to one or more machines. As in several other disciplines, e.g., operating room planning (see Cardoen et al. (2010)), we use the idea of 'three-fields' $(\alpha|\beta|\gamma)$ to provide a detailed description of a sports timetabling problem. The first field, α, determines the tournament format, the compactness of the timetable, and the required symmetry. The second field, β, lists around 20 constraint types partitioned into five classes that are able to model the vast majority of the constraints in the literature. Lastly, the γ-field refers to the objective in use. Sections 3.1 to 3.3 explain how each of these fields describe certain properties of a single-league timetabling problem. Next, Section 3.4 explains how to use the three-field notation in single- and multi-league settings.

3.1. Field α: competition format

The α-field features three parameters to represent the tournament format (α_1), the compactness (α_2), and the symmetry properties of the timetable (α_3). First, the α_1 parameter denotes a k-round-robin tournament with 'kRR', a k-bipartite round-robin tournament with 'kBRR', and all other non-round-robin tournaments with 'NRR'. Second, the parameter α_2 describes

Table 2: An illustration of different symmetry schemes for the games of team 2 in Table 1.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
Mirrored	(1,2)	(2,5)	(2,4)	(2,3)	(6,2)	(2,1)	(5,2)	(4,2)	(3,2)	(2,6)
Inverse	(1,2)	(2,5)	(2,4)	(2,3)	(6,2)	(2,6)	(3,2)	(4,2)	(5,2)	(2,1)
English	(1,2)	(2,5)	(2,4)	(2,3)	(6,2)	(2,6)	(2,1)	(5,2)	(4,2)	(3,2)
French	(1,2)	(2,5)	(2,4)	(2,3)	(6,2)	(5,2)	(4,2)	(3,2)	(2,6)	(2,1)

the compactness of the tournament: time-constrained tournaments are denoted by the value 'C', time-relaxed tournaments by the value 'R'. Finally, the third parameter α_3 denotes the symmetry of the timetable. By default, we assume that a tournament does not require any symmetry at all ('$\emptyset$'). However, in case of a kRR or kBRR tournament with $k > 1$, the α_3-part considers the following symmetry structures: phased ('P'), mirrored ('M'), inverted ('I'), English ('E'), or French ('F').

3.2. Field β: constraints

Sports timetables need to satisfy a usually large set of constraints C, which is partitioned into hard constraints C_{hard} and soft constraints C_{soft}. Hard constraints represent fundamental properties of the timetable that can never be violated. Soft constraints, in contrast, rather represent preferences that should be satisfied whenever possible. As in high school timetabling (see Post et al. (2012)), the validation of each constraint $c \in C$ results in a vector D_c of n_c integral numbers, called the deviation vector $D_c = [d_1\ d_2\ \ldots\ d_{n_c}]$. If a constraint is satisfied, all elements of its deviation vector are equal to zero. Contrarily, the deviation vector of a violated constraint contains one or more strictly positive elements. Apart from a description of how to calculate the deviation vector, each constraint features a cost function f_c and weight w_c. A violated constraint triggers a penalty $p_c = w_c f_c(D_c)$, equal to a weighted mapping of its deviation vector by its cost function. Our notation provides a total of five different cost functions. To begin, the sum function simply sums over all elements in the deviation vector (1). Similarly, the square-sum function (2) squares the sum of the deviations, and the sum-square function sums over all squared deviations (3). Finally, the min function (4) and max function (5) respectively output the smallest and largest element of the deviation vector.

$$f_c^{\text{sum}}(D_c) = \sum_{1\le i\le n_c} d_i \tag{1}$$

$$f_c^{\text{sq-sum}}(D_c) = (\sum_{1\le i\le n_c} d_i)^2 \tag{2}$$

$$f_c^{\text{sum-sq}}(D_c) = \sum_{1\le i\le n_c} d_i^2 \tag{3}$$

$$f_c^{\min}(D_c) = \min_{1\le i\le n_c} d_i \tag{4}$$

$$f_c^{\max}(D_c) = \max_{1\le i\le n_c} d_i \tag{5}$$

The evaluation of a timetable consists in the validation of all constraints and results in an infeasibility value and an objective value. The infeasibility value sums over all violated hard constraint penalties: a timetable is feasible if and only if it has an infeasibility value of zero. If there are soft constraints, the objective value sums over all violated soft constraint penalties; in the absence of soft constraints, other objectives can be specified (see Section 3.3).

The β-field categorizes the constraints from the literature into five different constraint classes. The remainder of this section discusses each constraint class and describes each member constraint together with its deviation vector. Some of these constraints offer a set of options, indicated using braces, to specify different variants of the constraint. In this case, the description of the deviation vector always assumes that the first option within braces is chosen. For all other options, we assume that the reader can adapt the description accordingly. In order to reduce the ambiguity between constraints, some constraints also impose restrictions on the (combination of) specified options.

3.2.1. Capacity constraints

Capacity constraints force a team to play home or away and regulate the total number of games played by a team or team group.

`CA1` Each team in team group T plays at least k_{min} and at most k_{max} {home games, away games, games} in time group S.
Each team in T triggers a deviation equal to the number of home games in S less than k_{min} or more than k_{max}.
$\forall i \in T : d_i = \max(k_{\min} - \sum_{j\in U}\sum_{s\in S} x_{i,j,s};\ \sum_{j\in U}\sum_{s\in S} x_{i,j,s} - k_{\max};0)$

`CA2` Each team in team group T_1 plays at least k_{min} and at most k_{max} {home games, away games, games} against {teams, each team} in team group T_2 in time group S.
Ambiguity breaking with GA1: $|T_1| > 1 \lor |T_2| > 1$.
Each team in T_1 triggers a deviation equal to the number of home games against teams in T_2 in S less than k_{min} or more than k_{max}.
$\forall i \in T_1 : d_i = \max(k_{\min} - \sum_{j\in T_2}\sum_{s\in S} x_{i,j,s};\ \sum_{j\in T_2}\sum_{s\in S} x_{i,j,s} - k_{\max};0)$

`CA3` Each team in team group T_1 plays at least k_{min} and at most k_{max} {home games, away games, games} against teams in team group T_2 in each sequence of k {time slots, games}.
Each team in T_1 triggers a deviation equal to the sum of the number of home games against teams in T_2 less than k_{min} or more than k_{max} for each sequence of k time slots.
$\forall i \in T_1 : d_i = \sum_{l=1}^{|P|-k+1} (\max(k_{\min} - \sum_{j\in T_2}\sum_{s=l}^{l+k-1} x_{i,j,s};$
$\sum_{j\in T_2}\sum_{s=l}^{l+k-1} x_{i,j,s} - k_{\max};0))$

`CA4` Teams in team group T_1 play at least k_{min} and at most k_{max} {home games, away games, games} against teams in team

group T_2 in {time group, each time slot of time group} S.
Ambiguity breaking with CA2: $|T_1| > 1$.
Time group S triggers a deviation equal to the number of games with a home team in T_1 and a team in T_2 less than k_{min} or more than k_{max}.
$d = \max(k_{\min} - \sum_{i \in T_1} \sum_{j \in T_2} \sum_{s \in S} x_{i,j,s}; \sum_{i \in T_1} \sum_{j \in T_2} \sum_{s \in S} x_{i,j,s} - k_{\max}; 0)$

CA5 Each team in team group T_1 plays at least k_{min} and at most k_{max} away games against a team in team group T_2 when it consecutively plays away during time group S.
Each team in T_1 triggers a deviation equal to the sum of the number of away games against teams in T_2 less than k_{min} or more than k_{max} for each sequence of away games in S.
$\forall i \in T_1 : d_i = \sum_{s_1 \in S} \sum_{s_2 \in S : s_2 > s_1} a_{i,s_1,s_2} \max(k_{min} - \sum_{j \in T_2} \sum_{s_3 = s_1}^{s_2} x_{j,i,s_3}; \sum_{j \in T_2} \sum_{s_3 = s_1}^{s_2} x_{j,i,s_3} - k_{max}; 0)$

Constraint CA1 is of fundamental use in sports timetabling to model 'place constraints' (Rasmussen and Trick, 2008) that forbid a team to play a home game, away game, or any game in a given time slot. Constraint CA1 can also help to balance the home-away status of games over time and teams. As an example, most phased tournaments regulate the number of home games each team plays in each interval. However, CA1 cannot be used to limit the difference in the number of home games played by a pair of teams, this can be accomplished by using FA2 (see Section 3.2.4) instead. As another example, when the home team receives ticket revenues, teams often request to play at least a given number of home games during the most lucrative time slots. Finally, CA1 can be used to model rounds, i.e., sets of time slots, usually weekends, during each of which a team can play at most one game.

Constraint CA2 can model 'top team and bottom team constraints' (Rasmussen and Trick, 2008) that prohibit bottom teams from playing all initial games against top teams. In odd- or non-round-robin tournaments, it is not only important to control for the timing of home games, but also for the opponents. In this regard, teams are often partitioned into different strength groups for which limits on the number of home games against each group are imposed. Constraint CA2 can also be used to model a kRR or kBRR between a subset of teams, or to put limits on the number of encounters in an NRR. Finally, CA2 can be used as a soft constraint if a kRR is preferably, but not necessarily phased. We point out that constraint CA2 actually generalizes constraint CA1. However, since CA1 and CA2 are of such a fundamental use in sports timetabling (see Section 4), a considerable amount of meaningful information would be lost when merging these two constraints.

Constraint CA3 can restrict a team to play at most one game in any sequence of k time slots so that this team has a guaranteed rest time (Suksompong, 2016) of $k-1$ time slots between any two games. Depending on whether byes are important, CA3 has different modes to limit the length of home stands and away trips. Similarly, CA3 can limit the total number of away breaks that occur when a team plays consecutively away against two opponents from a different geographical team group (e.g., Recalde et al. (2013)). However, CA3 should never be used to limit the total number of breaks; instead, this can be achieved using BR1 and BR2 (see Section 3.2.3). Finally, CA3 can model a *group changing* timetable in which teams are partitioned into g strength groups and it is required that no team plays consecutively against a team from the same team group. Similarly, a timetable is *group balanced* if no team plays more than once against teams of the same group within g consecutive games (Briskorn, 2009).

In contrast to CA2 and CA3 that define restrictions for each team in T_1, CA4 considers T_1 as a single entity. This constraint can therefore limit the number of games played between top teams in the same team group. As an example, to increase television viewership, Brazilian broadcasters request to balance football games between top teams over certain season intervals (see Ribeiro and Urrutia (2007b)). Constraint CA4 can also model asynchronous tournaments (e.g., Suksompong (2016)) that allow at most one game per time slot, or 'derby time slots' that only allow games between teams located in the same geographical area (e.g., Larson and Johansson (2014)). However, constraints that forbid a game between two specific teams in a given time slot must be classified with GA1 (see Section 3.2.2).

Finally, CA5 can forbid unreasonable road trips when teams play a series of consecutive away games without returning home. As an example, Chilean football teams cannot play in remote places during road trips (Durán et al., 2007, 2012). However, note that constraints regulating the total number of road trips must be classified with BR4 (see Section 3.2.3).

3.2.2. Game constraints

Game constraints enforce or forbid specific assignments of a game to time slots.

GA1 At least $k_{\min}$ and at most $k_{\max}$ games from $G = \{(i_1, j_1), (i_2, j_2), \dots\}$ take place in time group S.
Time group S triggers a deviation equal to the number of games in G less than $k_{\min}$ or more than $k_{\max}$.
$d = \min(k_{\min} - \sum_{(i,j) \in G} \sum_{s \in S} x_{i,j,s}; \sum_{(i,j) \in G} \sum_{s \in S} x_{i,j,s} - k_{\max}; 0)$

GA2 If a team from team group T_1 plays a {home game, away game, game} against a team from team group T_2 in time group S_1, then a team from team group T_3 {plays, does not play} a {home game, away game, game} against a team from team group T_4 in time group S_2.
Time group S_2 triggers a deviation of 1 if a team in T_1 plays a home game against a team in T_2 in S_1 but no team in T_3 plays a home game against a team in T_4 in S_2, and 0 otherwise.
$d = \max(\min(\sum_{i \in T_1} \sum_{j \in T_2} \sum_{s \in S_1} x_{i,j,s}, 1) - \min(\sum_{k \in T_3} \sum_{l \in T_4} \sum_{s \in S_2} x_{k,l,s}; 1); 0)$

Constraint GA1 deals with fixed and forbidden game to time slot assignments and is of fundamental use in sports timetabling. Examples include the police that forbid to play high risk games during time slots in which other major events are planned, and broadcasters that request at least one 'top game' or 'classic game' in each televised time slot. Constraint GA2 is able to model conditional requirements between games. As an example, Chilean television broadcasters request that no top team plays in the North when another top team plays in the South (Durán et al., 2007). Likewise, the timetable in Nurmi et al. (2015) must include a weekend where teams playing each other on Friday also play each other on Saturday.

3.2.3. Break constraints

Break constraints regulate the frequency and timing of breaks in a competition. A team has a home (away) break when it plays two consecutive home (away) games, no matter how many byes this team has between the two games.

`BR1` Each team in team group T has {exactly, no more than} k {home breaks, away breaks, breaks} in time group S.
Each team in T triggers a deviation equal to the difference in the sum of home breaks in S and k.
$\forall i \in T : d_i = |k - \sum_{s \in S} b_{i,s} h_{i,s}|$

`BR2` The sum over all {home breaks, away breaks, breaks} of teams in team group T is {exactly, no more than} k in time group S.
Ambiguity breaking with BR1: $|T| > 1 \wedge k > 0$.
Team group T triggers a deviation equal to the difference in the sum of home breaks in S and k.
$d = |k - \sum_{i \in T} \sum_{s \in S} b_{i,s} h_{i,s}|$

`BR3` Each pair of teams in team group T has a difference in {home breaks, away breaks, breaks} that is not larger than k.
Each pair of teams in T triggers a deviation equal to the difference in the number of home breaks more than k.
$\forall i, j \in T, i < j : d_{i,j} = \max(|\sum_{s \in P} b_{i,s} h_{i,s} - \sum_{s \in P} b_{j,s} h_{j,s}| - k, 0)$

`BR4` Each team in team group T has at least k road trips in time group S.
Each team in T triggers a deviation equal to the number of road trips in S less than k.
$\forall i \in T : d_i = \max(k - \sum_{s_1 \in S} \sum_{\substack{s_2 \in S: \\ s_2 > s_1}} a_{i,s_1,s_2}; 0)$

Breaks usually are undesired since they have an adverse impact on game attendance (see Forrest and Simmons (2006)) and they can be perceived as unfair due to the home-away effect (e.g., Pollard and Pollard (2005)). For this reason, BR1 can forbid breaks at the beginning or end of the season, or can limit the total number of breaks per team. Constraint BR2, on the other hand, can limit the total number of breaks in a competition. Alternatively, organizers may use BR3 to enforce break equitable timetables in which all teams have the same number of breaks, although the total number of breaks then possibly increases. In case that the distance between the venues of each team pair is constant, Urrutia and Ribeiro (2006) show that the minimization of distance is equivalent with break maximization. In order to reduce travel distance, some teams might therefore prefer to make at least a given number of road trips. A combination of BR4 and CA5 can handle the situation in which the composition of road trips is also important.

3.2.4. Fairness and attractiveness constraints

The following constraints increase the fairness or attractiveness of competitions.

`FA1` Each team in team group T has a difference in played home and away games that is not larger than k after each time slot in S.
Each team in T triggers a deviation equal to the largest difference in played home and away games more than k over all time slots in S.
$\forall i \in T : d_i = \max_{s \in S}(|\sum_{j \in T} \sum_{1 \leq p \leq s} (x_{i,j,p} - x_{j,i,p})| - k; 0)$

`FA2` Each pair of teams in team group T has a difference in played {home games, away games, games} that is not larger than k after each time slot in S.
Each pair of teams in T triggers a deviation equal to the largest difference in played home games more than k over all time slots in S.
$\forall i, j \in T, i < j : d_{i,j} = \max_{s \in S}(|\sum_{t \in T} \sum_{1 \leq p \leq s} (x_{i,t,p} - x_{j,t,p})| - k; 0)$

`FA3` Each pair of teams in team group T plays each other at home and in turn away.
Each pair of teams in team group T triggers a deviation equal to the total number of times the two teams play consecutively with the same home-away assignment.
$\forall i, j \in T, i < j : d_{i,j} = \sum_{k=1}^{g_{i,j}+g_{j,i}-1} (1 - |h_{i,j,k} - h_{i,j,k+1}|)$

`FA4` Team group T has a {weighted coe, coe} value of at most k.
Team group T triggers a deviation equal to the weighted coe-value more than k.
$d = \max(\sum_{i \in T} \sum_{j \in T} w_{i,j} c_{i,j}^2 - k; 0)$

`FA5` The total distance traveled by all teams in team group T during time group S is at most k.
Team group T triggers a penalty equal to the total distance traveled more than k.
$d = \max(\sum_{i \in T} \sum_{s \in S} e_{i,s} - k; 0)$

`FA6` The total cost associated with all games played during time group S is at most k.
Time group S triggers a penalty equal to the total cost more than k.
$d = \max(\sum_{i \in T} \sum_{j \in T} \sum_{s \in S} c_{i,j,s} x_{i,j,s} - k; 0)$

Over the years, various metrics have been developed to increase the fairness of sports timetables. As an example, Knust and von Thaden (2006) call a timetable balanced if the difference in played home and away games for each team is at most 1 at the end of the season. Nurmi et al. (2010) generalize this measure to k-balancedness which requires the difference in played home and away games to be smaller than k at any point in time (FA1). In phased tournaments without additional symmetry requirements, FA1 may also enforce (k-)balancedness at the end of each phase. However, we use constraint CA1 to accomplish that a team plays a given number of home games in a certain time group. To increase the accuracy of the competition rankings, Goossens and Spieksma (2011) call a timetable k-ranking-balanced if, after each time slot, 'the difference between the number of home games played by any two teams up till then is at most k' (FA2). Related to this, Suksompong (2016) proposes the games-played difference index that measures 'the minimum integer k such that at any point in the timetable, the difference between the number of games played by any two teams is at most k' (FA2). In tournaments in which some teams meet more than twice, constraint FA3 enforces that each pair of teams meet each other at home and in turn away (e.g., Carlsson et al. (2017)). Constraint FA4 can be used to model a coe-balanced timetable in which the total carryover effects value is minimum. Similarly, FA5 can enforce distance minimal timetables but can also more fairly distribute the total travel distance among teams. Finally, FA6 limits the total cost associated with the game assignments in a timetable. As an example, some competitions try to minimize the distance traveled by the fans over holiday periods (e.g., Kendall (2008); Westphal (2014); Cocchi et al. (2018)). This is different from FA5 since we assume that fans of a team immediately return back to the home venue of their team. Constraint FA6 can model this by setting a cost of $c_{i,j,s} = d_{i,j}$ for all $s \in S$ and $i, j \in T$.

3.2.5. Separation constraints

Finally, separation constraints regulate the number of time slots between consecutive games involving the same teams and regulate the symmetry of the timetable.

SE1 Each pair of teams in team group T has at least k {time slots, games} between two consecutive mutual games.
Each pair of teams in T triggers a deviation equal to the sum of the number of time slots less than k for all consecutive mutual games.
$\forall i, j \in T, i < j : d_{i,j} = \sum_{s_1 \in P} \sum_{s_1+1 \leq s_2 \leq |P|} y_{i,j,s_1,s_2} \max(k - (s_2 - s_1); 0)$

SE2 If a pair of teams in team group T meets in one time slot of a pair of time slots in $Q = \{\{s_1, s_2\}, \{s_3, s_4\}, \dots\}$, it also meets in the other time slot.
Each pair of time slots in Q triggers a deviation equal to the number of pairs of teams that play in one time slot but not in the other time slot.
$\forall \{s_1, s_2\} \in Q : d_{\{s_1,s_2\}} = \sum_{\{i,j\} \in T} |(x_{i,j,s_1} + x_{j,i,s_1}) - (x_{i,j,s_2} + x_{j,i,s_2})|$

Organizers may request that two games with the same opponents are separated by at least a given number of time slots (SE1). If a separation of at least one time slot is required, this constraint is often called the 'no-repeater' constraint (Easton et al., 2001). Another way to enforce this separation in a time-constrained kRR with an even number of teams is to use a mirrored symmetry structure (α_3 = 'M') resulting in a maximum achievable separation for all games (Goossens and Spieksma, 2012). Constraint SE2 can be used if a symmetry structure other than those in the α_3-field is required (see e.g., Nemhauser and Trick (1998)), or when a symmetry structure should be modeled as a soft constraint as it is preferred but not strictly required.

Table 3: Mathematical description of the different objectives functions.

Objective	Symbol	Calculation
Minimum (weighted) break	BR	$\sum_{i \in U} \sum_{s \in P} v_{i,s} b_{i,s}$
Minimum travel	TR	$\sum_{i \in U} \sum_{s \in P} e_{i,s}$
Minimum cost	CR	$\sum_{i \in U} \sum_{j \in U \setminus i} \sum_{s \in S} x_{i,j,s} c_{i,j,s}$
Minimum (weighted) coe-value	CO	$\sum_{i \in U} \sum_{j \in U} w_{i,j} c_{i,j}^2$
Minimum soft constraint	SC	$\sum_{c \in C_{\text{soft}}} p_c$

3.3. Field γ: objective

As explained in Section 3.2, every timetable has an infeasibility value and an objective value. Where the infeasibility value of a timetable is completely determined by the sum of the penalty values of violated hard constraints, the γ-field describes how to calculate the objective value. If no objective is provided ('$\emptyset$'), the problem reduces to a constraint satisfaction problem in which the sole purpose is to find a timetable respecting all hard constraints.

Similarly, the minimum break objective (BR) minimizes the total number of breaks in the competition. In the weighted break minimization problem, breaks are additionally weighted (see, e.g., Durán et al. (2017)). For a 1RR timetable with an even number of teams and no additional requirements, de Werra (1981) proves that the minimum number of breaks equals $|U| - 2$. Moreover, if the number of teams is odd or if the competition is time-relaxed, Fronček and Meszka (2005) show that a 1RR timetable exists without any breaks. If there is no need for the timetable to be phased, the minimum number of breaks does not increase in a kRR when k increases (e.g., Goossens and Spieksma (2011)). Contrarily, if the timetable needs to be phased and a minimal separation between mutual games is required, the construction of a break minimal timetable becomes less obvious (see Zeng and Mizuno (2012)).

In many large countries, it is necessary to reduce the amount of travel by organizing road trips where teams play a series of consecutive away games without returning home. The minimum distance objective (TR) tries to minimize the total distance traveled by all teams in the competition and can be used to model the traveling tournament problem (Easton et al., 2001).

In several settings, we can associate a cost or revenue to play a given game in a particular time slot. As an example, attendance estimates can be used to maximize revenues. Likewise, costs can model venue rental prices if venues are owned by public agencies to which teams have to pay a fee if they play a home match in that stadium (see e.g., Briskorn and Drexl (2009a)). The minimal cost problem (CR) constructs cost-minimal timetables (and is equivalent to the maximum revenue problem).

As a response to managers complaining to be disadvantaged by the timetable because of carryover effects, Goossens and Spieksma (2012) analyzed more than 10,000 Belgian football games but found no evidence of unbalanced carryover effects impacting team performance. Notwithstanding, the construction of timetables with a minimum (weighted) coe-value (CO) remains a popular subject, even in the Belgian football competition (Goossens and Spieksma, 2009).

In contrast to the previous objectives that assume that all constraints are hard, the soft-constraints objective (SC) assumes that the problem features one or more soft constraints, and minimizes the total penalty resulting from violated soft constraints while still respecting all hard constraints. Note that all of the previous objectives can be expressed in the form of constraints: use BR2 to limit the total number of breaks, FA5 to limit the total travel distance, FA6 to limit the total costs or guarantee a certain revenue, and FA4 to limit the carryover effects value. Table 3 gives an overview of the different objectives and their mathematical description.

3.4. Delimiters

The previous sections explained how to use each field parameter in isolation but did not exemplify how to combine the different parameters into a single meaningful string. Hereto, Table 4 provides an overview of the different delimiters and superscripts used to further annotate the three-field notation. First, we use a vertical bar to separate the three different fields from each other. Within each field, we then use a comma to separate information belonging to different parameters within the same field. If the timetabling problem features soft constraints, i.e., we selected the soft-constraints objective, we need a delimiter to distinguish the soft constraints from hard constraints. If the constraint appears both as a hard constraint and soft constraint, we annotate its label with the letters 'H,S'; otherwise we add the superscript 'H' for hard or 'S' for soft.

Until now, we only explained how to use the three-field notation for single-league problems. In case the problem features multiple leagues, we first apply the notation for all leagues individually and separate the information in parentheses. If there are inter-league constraints that cover teams in more than one given league, the constraint label is added to the notation of each of the given leagues. Next, we concatenate the notations of the individual leagues by using a plus operator. To condense information, the notation allows to join the notation of similar leagues; a superscript indicates how many different leagues follow the format in parentheses. An overview of the three-field notation and all its different field, parameters, and values is given in Figure 2.

We use the three-field notation outlined in this section on two different abstraction levels. First, Section 4 operates on the highest level of abstraction and applies the three-field notation to generate a classification string for various sports timetabling problems published in the literature. Despite being compact, the classification string may hide problem specific characteristics: it does for example not distinguish between the different constraint options indicated in braces. If a more fine-grained description is needed, we refer the reader to Section 5 which

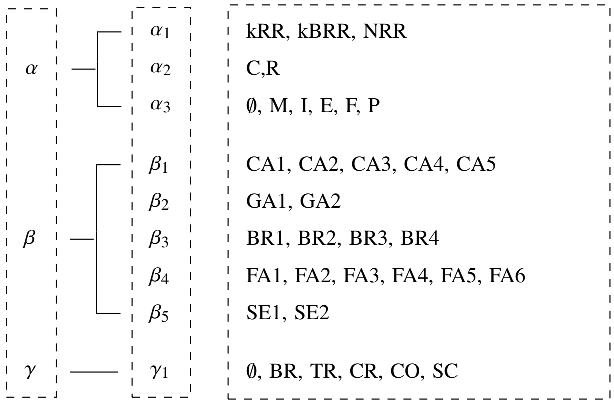

Figure 2: Overview of the three-field notation for sports timetabling.

expresses sports timetabling problems in full detail by using xml files which structure is based on the three-field notation.

4. Classification on sports timetabling problems

To get a thorough understanding of the typical structure of (round-robin) sports timetabling problems, Section 4.1 provides a problem-driven classification of the sports timetabling literature. Next, Section 4.2 discusses considerations, challenges, and trade-offs that arose during the development of the three-field notation that had to be capable of carrying out this classification.

4.1. A problem-driven classification

Several classifications on (round-robin) sports timetabling have previously been published in highly ranked journals. Drexl and Knust (2007) give an excellent overview of graph-based models for sports timetabling. Rasmussen and Trick (2008) list important sports timetabling contributions concerning break and travel optimization and clarify various timetabling notations. Despite their substantial effort to unify terminology, the sports timetabling notation used in the literature remains highly inconsistent. Kendall et al. (2010) give an overview of important sports timetabling problems, methodologies, and applications. Finally, Ribeiro (2012) lists a number of sports timetabling problems and examines different mathematical formulations. Although these classification papers give an excellent overview of sports timetabling contributions over time, they do not enable researchers to extract detailed information about the tournament characteristics and the constraints or objective in use. At best, these classifications inform the reader about the objective and one or two specific constraints. This is problematic since the popularity of particular tournament formats, constraints, and objectives remains unidentified.

To direct future research efforts towards the most common problem instance configurations, Table 4.1 employs the three-field notation from Section 3 to classify 61 real-life sports timetabling problems that previously appeared in the operations research literature (based on title and abstract selection

Table 4: Summary on the use of delimiters and superscripts in the three-field notation.

Delimiter	Function	Example
$\mid$	Field delimiter	$\alpha \mid \beta \mid \gamma$
,	Parameter delimiter	2RR,C,M $\mid \beta \mid \gamma$
CA1$^{\mathrm{H,S}}$	Hard versus Soft constraint	$\alpha \mid$ CA1$^{\mathrm{H,S}}$, CA2$^{\mathrm{H}}$, CA3$^{\mathrm{S}}$ $\mid$ SC
$()^l$	Delimiter to group the problem notation for l coherent leagues	$(\alpha \mid \beta \mid \gamma)^l$
+	Delimiter to concatenate the notation of different coherent leagues	$(\alpha \mid \beta \mid \gamma)^{l_1} + (\alpha \mid \beta \mid \gamma)^{l_2}$

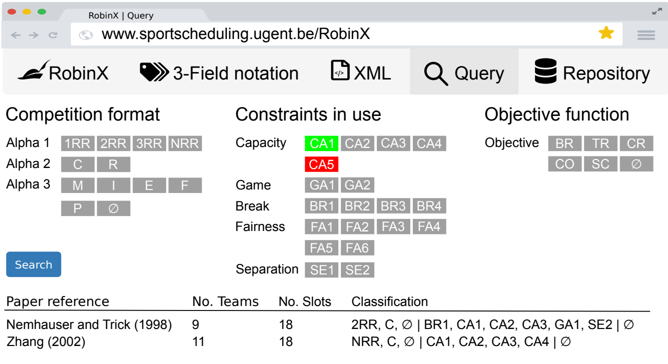

Figure 3: Illustration of the online query tool. Users color buttons green or red to respectively select instances that have or do not have specific parameter values.

from (Knust, 2018)). Note that Table 4.1 gives a classification on sports timetabling *problems*, rather than research papers. Consequently, if a single paper introduces multiple sports timetabling problems, the same paper reference may appear more than once, each time with a different three-field notation.

For each problem, a list with the specific constraints as originally described in the paper is publicly available on the website devoted to this project (see Van Bulck et al. (2018b)). The website also offers a query tool in which users can select and deselect problem features to get a list with all matching problems from a database that additionally contains the classification for several theoretical problem instances (e.g., the constrained minimum break problem (Rasmussen and Trick, 2007), and the tournament scheduling problem with absences (Schauz, 2016)). Confronted with a specific sports timetabling problem, this tool should enable users to easily identify the most relevant part of the literature (see Figure 3).

Figure 4 illustrates the frequency of each of the constraints as they appear in Table 4.1. From this figure, it is apparent that capacity constraints are by far the most popular. Also popular is to limit the total number of breaks per team in a given set of time slots (BR1), to limit the total number of breaks (BR2), to forbid specific game assignments (GA1), and to require a minimal separation between common games (SE1).

4.2. Discussion

Several considerations came to mind during the development of the three-field notation outlined in Section 3. A first consideration is related to the level of abstraction used in the constraint formulation (for a discussion, see Kingston (2018)). Too specific constraints result in an explosion of the total number of constraints and are likely unable to handle new instances with slightly different requirements. Too general constraints, on the other hand, lose their expressive power by hiding too much information so that problem specific knowledge can no longer be exploited. In the most extreme case, one generic constraint could be used to classify all timetabling requirements leaving no room for ambiguity. For instance, some early high school timetabling notations proposed specification languages that allowed users to define any constraint that can be expressed via a function or logic expression (e.g., Kingston (2001)). Despite the inherent flexibility, these notations were not adopted by the research community because of the high specification burden. Related to this, a now widely accepted high school timetabling format (XHSTT) defines meetings as abstract *events* that need to be assigned *resources* thereby considering constraints on an event-resource level (see Post et al. (2012); Kingston et al. (2018)). Following this approach, a game could be modeled as an event that requires a time slot resource and two team resources, one with a 'home role' and one with an 'away role'. Although our three-field notation carried over many ideas from XHSTT, such as the profound idea to express constraints in function of any arbitrary set of time slots or teams, it does not use the idea of events and resource-based constraint formulations. The reason is that the additional layer of abstraction would likely transform the three-field notation into a meaningless string of tokens that hides fundamental problem characteristics. Instead, the formulation of constraints is based on the long list of real-life constraints encountered during the problem classification. In other words, our notation rather resembles the way how practitioners tend to structure data (domain specific structure) than how solvers treat requirements via variables and resources (solver specific structure) (see Post et al. (2014)).

As a result of the specificity-generality trade-off, not all ambiguity was resolved and some complicated cases exist in which it may not be so clear how to classify a given constraint. As an example, suppose that a top game is defined as a game between two teams from a given set of top teams. To maximize television viewership, it is often imposed that no top game takes place during the first two time slots. To classify this constraint, it is not so clear whether we should use CA2, CA4, or GA1; whenever this occurs we advise researchers to choose the constraint that most closely matches the description. For example, we recommend CA2 for the description given above. However, we advise CA4 if the constraint is formulated in the form 'at most one top team can play at home against another top team in the first and second time slot'. We advise GA1 in case a more arbitrary set of games is involved.

Table 5: Three-field classification of literature on sports timetabling.

Paper reference	**Description**
Constraint satisfaction	
Nemhauser and Trick (1998)	2RR, C, ∅ | BR1, CA1, CA2, CA3, GA1, SE2 | ∅
Zhang (2002)	NRR, C, ∅ | CA1, CA2, CA3, CA4 | ∅
Van Voorhis (2005)	NRR, R, ∅ | CA1, CA2, CA3, GA1, SE1 | ∅
Van Voorhis (2005)	NRR, R, ∅ | CA1, CA2, CA3, GA1, SE1 | ∅
Kostuk and Willoughby (2012)	NRR, R, ∅| BR1, CA1, CA2, CA3, CA4, GA1 | ∅
Break optimization	
de Werra (1985)	1RR, C, ∅| BR3, CA2, CA3 | BR
de Werra (1985)	1RR, C, ∅| BR1, CA3, CA4 | BR
Della Croce and Oliveri (2006)	2RR, C, M | CA1,CA2, CA3, CA4| BR
Saur et al. (2012)	1RR, C, ∅ | BR1, CA1, CA2, CA3, CA4 | BR
Durán et al. (2017)	2RR, C, F| CA1, CA3 | BR
Travel optimization	
Ball and Webster (1977)	2RR, R, P | GA2, CA3, | TR
Bean and Birge (1980)	NRR, R, ∅ | CA1, CA3 | TR
Russell and Leung (1994)	NRR, C, ∅ | CA2, CA3 | TR
Easton et al. (2001)	2RR, C, ∅ | CA3, SE1 | TR
Ribeiro and Urrutia (2007a)	2RR, C, M| CA3, SE1| TR
Bao and Trick (2010)	2RR, R, ∅ | CA3, SE1| TR
Hoshino and Kawarabayashi (2011a)	8RR, C, P| CA3, FA1, SE1| TR
Hoshino and Kawarabayashi (2011b)	2BRR, C, ∅ | CA3, SE1| TR
Bonomo et al. (2012)	2RR, C, M | BR1, CA1, CA3 | TR
Cost and revenue optimization	
Durán et al. (2007)	1RR, C, ∅ | BR1, CA1, CA2, CA3, CA4, CA5, GA1, GA2 | CR
Briskorn and Drexl (2009b)	1RR, C, ∅ | BR1, BR2, CA1, CA3, CA4, GA1| CR
Fiallos et al. (2010)	2RR, C, M | BR1, CA2, CA3, CA4 | CR
Durán et al. (2012)	4RR, C, M | BR1, BR4, CA1, CA3, CA4, CA5, FA1, GA1 | CR
Carryover effects	
Guedes and Ribeiro (2011)	1RR, C, ∅ | | CO
Günneç and Demir (2018)	2RR, C, M | BR1 | CO
Soft constraint	
Ferland and Fleurent (1991)	NRR, R, ∅ | CA1^{H,S}, CA2^{H}, CA3^{H,S}, FA5^{S}, GA2^{H}, SE1^{H} | SC
Schreuder (1992)	2RR, C, M | BR2^{S}, CA1^{H,S}, CA4^{S} | SC
Costa (1994)	NRR, R, ∅ | CA1^{H,S}, CA2^{H}, CA3^{H,S}, FA5^{S}, GA2^{H}, SE1^{H} | SC
Wright (1994)	NRR, R, ∅ | CA1^{H}, CA2^{H}, CA3^{S}, CA4^{H,S}, FA5^{S}, GA1^{H} | SC
Dinitz and Froncek (2000)	NRR, C, ∅ | BR1^{H}, BR2^{S}, CA1^{H}, CA2^{H}, CA3^{H}, CA4^{H}, SE1^{S} | SC
Schönberger et al. (2000)	1RR, R, ∅ | CA1^{H}, CA2^{H}, CA3^{S}, FA1^{H} | SC
Frončck (2001)	2RR, C, P| BR1^{H}, BR2^{H,S}, CA2^{H}, CA3^{H}, CA4^{H}, GA2^{H,S}, SE2^{S} | SC
Easton (2003)	2RR, C, ∅ | BR1^{H}, CA1^{H}, CA2^{H,S}, CA3^{H}, CA5^{H}, SE1^{H} | SC
Schönberger et al. (2004)	2RR, R, P| CA1^{H,S}, CA3^{S} | SC
Wright (2005)	4RR, R, ∅ | CA1^{S}, CA2^{H,S}, CA4^{S}, FA1^{S}, FA5^{S}, GA1^{S}, SE1^{S}, SE2^{S} | SC
Bartsch et al. (2006)	2RR, C, M | BR1^{H,S}, BR2^{H}, CA1^{H}, CA3^{H,S}, CA4^{H}, GA1^{H}, SE1^{H} | SC
Bartsch et al. (2006)	2RR, C, E | BR1^{H}, CA1^{H,S}, CA2^{H}, CA3^{S}, CA4^{H}, GA1^{H}, SE1^{H} | SC
Wright (2006)	2RR, C, ∅ | CA1^{S}, CA2^{S}, CA4^{S}, FA2^{S}, FA5^{S}, GA1^{S} | SC
Ribeiro and Urrutia (2007b)	2RR, C, M | BR1^{H}, BR2^{S}, CA1^{H}, CA2^{H}, CA4^{H,S}, FA1^{H} | SC
Rasmussen (2008)	3RR, C, P| BR1^{H,S}, BR2^{S}, CA1^{H}, CA2^{H,S}, CA3^{H}, CA4^{S}, GA1^{S}, SE1^{H} | SC
Goossens and Spieksma (2009)	2RR, C, M| BR1^{H}, BR2^{H}, CA1^{H,S}, CA2^{S}, CA3^{H,S}, CA4^{H,S}, GA1^{S} | SC
Kyngäs and Nurmi (2009a)	4RR, C, ∅ | BR1^{H}, BR4^{S}, CA1^{H,S}, CA3^{S}, FA1^{S}, GA1^{H}, SE1^{S} | SC
Kyngäs and Nurmi (2009b)	NRR, R, ∅ | BR1^{H}, BR4^{S}, CA1^{H,S}, CA3^{S}, CA4^{H,S}, FA1^{S}, FA2^{S}, GA1^{H}, SE1^{S} | SC
Knust (2010)	1RR, R, ∅ | BR2^{S}, CA1^{H,S}, CA2^{H}, CA3^{S}, FA1^{H}, GA1^{H} | SC
Lewis and Thompson (2011)	2RR, C, ∅ | CA1^{H}, CA2^{H,S}, CA3^{H,S}, CA4^{H}, SE1^{S} | SC
Hausken et al. (2012)	2RR, C, P| BR1^{H}, BR2^{S}, CA1^{S}, CA3^{S}, CA4^{H,S}, GA1^{H,S}, SE1^{H} | SC
Ribeiro and Urrutia (2012)	2RR, C, M | BR1^{H}, BR2^{H}, BR3^{H}, CA1^{H}, CA2^{H,S}, CA3^{H}, CA4^{H}, GA1^{H}, GA2^{H} | SC
Nurmi et al. (2013)	3RR, R,∅ | BR2^{S}, CA1^{H,S}, CA2^{H}, CA4^{H,S}, FA2^{S}, GA1^{H}, SE1^{S} | SC
Recalde et al. (2013)	2RR, C, I| BR1^{H,S}, CA2^{H}, CA3^{H,S}, CA4^{H} | SC
Larson and Johansson (2014)	NRR, C, ∅ | BR2^{H}, CA1^{H,S}, CA2^{H}, CA4^{H}, FA1^{H}, FA3^{H}, SE2^{H} | SC
Westphal (2014)	2RR, C, P | BR2^{S}, CA1^{H,S}, FA6^{S}, GA1^{S} | SC
Nurmi et al. (2015)	4RR, R, ∅ | BR1^{H}, BR2^{S}, BR4^{S}, CA1^{H,S}, CA3^{S}, CA4^{H,S}, FA1^{S}, FA2^{S}, FA3^{S}, GA1^{S}, GA2^{H}, SE1^{S} | SC
Kyngäs et al. (2017)	NRR, R, ∅ | BR1^{H}, BR2^{S}, CA1^{H}, CA2^{H}, CA3^{H,S}, CA4^{H}, FA5^{S}, GA1^{H}, SE1^{H} | SC
Cocchi et al. (2018)	2RR, C, M| BR1^{H,S}, BR2^{S}, CA1^{H,S}, CA2^{H,S}, CA3^{H}, CA4^{H}, FA1^{H}, FA6^{S}, GA1^{H,S} | SC
Van Bulck et al. (2019)	2RR, R, ∅ | CA1^{H}, CA3^{H,S}, SE1^{H} | SC
Multi-league	
de Werra et al. (1990)	(2RR,C,P | BR1^{H,S}, BR3^{S}, CA4^{H}, SE1^{H} | SC)^{1} + (NRR,C,∅ | BR1^{H,S}, BR3^{S}, CA2^{H}, CA4^{H} | SC)^{1}
Della Croce et al. (1999)	(1RR, R, ∅ | CA1, CA4 |∅)^{6}
Kendall (2008)	(NRR, C, ∅ | BR2^{H}, CA1^{H}, CA2^{H}, CA4^{H,S}, FA6^{H} | SC)^{4}
Grabau (2012)	(NRR, R, ∅ | CA1, CA2, CA3, CA4, GA2 | CR)^{8}
Burrows and Tuffley (2015)	(2RR, C, M | GA2^{S}, FA3^{H} | SC)^{1} + (1RR, C, ∅ | GA2^{S}, FA3^{H} | SC)^{1}
Schönberger (2017)	(2RR, R, ∅ | CA1^{H}, CA3^{S}, CA4^{H} | SC)^{3}

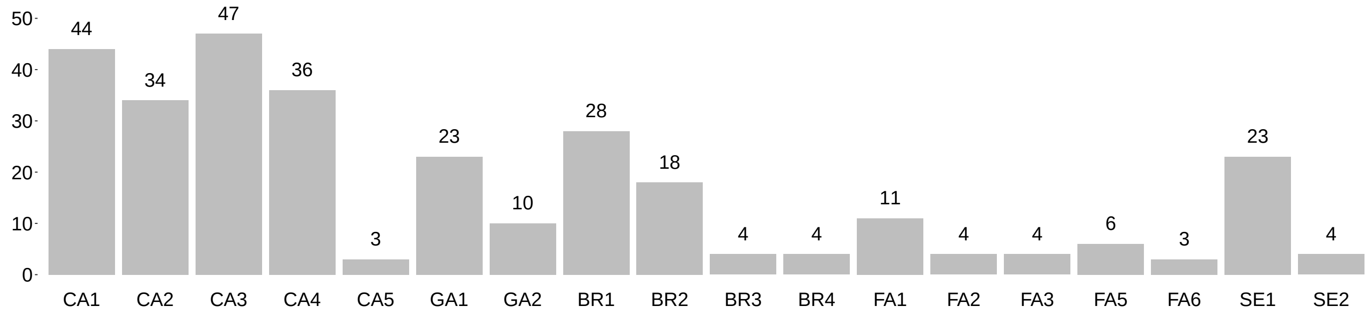

Figure 4: Frequency of the different constraints in real-life applications from Table 4.1. Constraint FA4 did not appear in the considered problems but is not removed from the three-field notation to be able to handle multi-objective settings.

The list of constraints and objectives outlined in Section 3 is able to classify most real-life sports timetabling applications published over the last five decades. Yet, as sports timetabling is an active research domain in which new problems are regularly proposed, an important question is how to adapt the framework when new problem properties arise. Clearly, we want to avoid that researchers independently decide what, for instance, the next CA6 will be as this will likely result in an uncontrolled explosion of the three-field notation. Neither is our intention to classify every possible timetabling constraint as some properties might be too problem specific. Instead, we propose that unclassifiable constraints are itemized in the problem description via a separate 'other constraints' list, and that researchers publish these constraints on the website devoted to this project (Van Bulck et al., 2018b). This allows the research community to be involved in the decisions of which extensions to include.

A final consideration relates to the assumption that a game is played in the venue of the home team. Although this assumption holds for the vast round-robin timetabling contributions in the literature, there are some real-life sports competitions that organize games on multiple neutral venues that do not belong to any team (e.g., McAloon et al. (1997); Urban and Russell (2003)). Apart from the time slot assignment, the timetable then also needs to assign a venue to each game. Note, however, that the use of neutral venues does not necessarily implicate that the home-away status becomes obsolete: the 'home team' in chess can move first, whereas the home team in baseball bats last. Although neutral venues do not belong to the scope of the proposed notation scheme, it can be modeled by formulating the problem in the same way as is done in balanced tournament design (see e.g., McAloon et al. (1997)). Within balanced tournament design, we are given a set of teams U, a set of venues V, and a set of time slots P with $|P| = 2|V|$. The task is then to organize a 1RR complemented with one game per team in such a way that each team plays twice on each venue but no two teams meet more than once on the same venue. To classify neutral venues, we replace the set of time slots with a new set of time slots P' that contains a time slot $s_{p,v}$ for each original time slot $p \in P$ and venue $v \in V$. This way, we automatically assign both a venue in V and time slot in P whenever we assign a game to a time slot in P'. In order to ensure that a team plays at most one game in a time slot $p \in P$, it suffices to define a constraint of type CA1 that states that a team can play at most one game in a time group consisting of all time slots $s_{p,v}$ with a venue $v \in V$. Similarly, a constraint of type CA4 can be used to model venue capacity.

5. A unified XML-file-based data format

Constructing sports timetables is a complex matter since, as illustrated in Section 4, real-life sports timetabling applications are typically highly constrained. The vast amount and variety of constraints and the lack of easily accessible benchmark problem instances induce that solution methods are often tested on just one or two specific instances. The construction of a problem instance repository, however, is not straightforward since there are dozens of constraints that need to be expressed in a standardized data format. Section 5.1 outlines the criteria for such a data format. Next, Section 5.2 proposes three XML standards for exchanging problem instances, solutions, and objective bounds in the field of sports timetabling. We use these standard to propose an instance data repository containing various sports timetabling instances. Finally, Section 5.3 proposes several tools and an online platform that facilitate the construction and validation of these XML files.

5.1. Design criteria and rationale for using XML

The main intention of our data format, also called information standard or medium, is to promote problem instance data sharing and reuse among different users and software applications. A first criterion is therefore that the format is open, human readable (i.e. no binary format), software and platform independent, and flexible enough to store the majority of the problem instances from the literature. Second, the format must aid users to structure input data in an uncomplicated and recognizable way, thereby minimizing the specification burden and maximizing the accessibility. Therefore, it is required that the format is methodology independent since the usage of the format should not be limited by the users' modeling capability (see also Chatfield et al. (2009)). Indeed, most of the timetabling constraints are easy to express in words but are hard to enforce within specific algorithms such as mathematical programming or metaheuristics. Third, the implementation effort for reading

```
<Instance>
  <MetaData> <InstanceName>Utopia</InstanceName> </MetaData
      >
  <Data> <Distances/> <Costs/> </Data>
  <Resources>
    <TeamGroups> <teamGroup id="1" name="All teams"/> </
         TeamGroups>
    <Teams> <team id="1" name="Team 1" teamGroups="1"/> </
         Teams>
    <TimeGroups>...</TimeGroups> <Slots> ... </Slots>
  </Resources>
  <Structure>
    <Format teamGroupIds="1">
      <numberRoundRobin>2</numberRoundRobin>
      <compactness>C</compactness> <symmetry>M</symmetry>
      <AdditionalGames/>
    </Format>
  </Structure>
  <Constraints>
    <CapacityConstraints>
        <CA1 teamGroup="1,2" min="1" max="2" mode="HOME"
             timeGroup="1,2,3" cost="1" type="SOFT" function
             ="SUM"/>
    </CapacityConstraints>
  </Constraints>
  <ObjectiveFunction> <Objective>SC</Objective> </
       ObjectiveFunction>
</Instance>
```

Figure 5: Example of the instance XML format.

data must be limited. Fourth, the format must be easy to extend if new timetabling formats, constraints, or objectives are introduced in the literature.

The extensible markup language (XML) meets all the above design criteria. Moreover, unlike other file formats such as JSON, XML has the advantage that many researchers are already familiar with this format because of its popular use in many other operations research disciplines such as high school timetabling (Post et al., 2012), nurse rostering (Kingston et al., 2018), multi-dimensional packing (Fekete and van der Veen, 2007), and supply chain modeling (Chatfield et al., 2009). For an introduction to XML and examples on the use of XML in operations research, we refer to Bradley (2003). The driving motivation behind XML is to separate data representation from data content. This separation is enforced by providing an XML language (also called tag set, vocabulary, document type, or standard) consisting of all recognized tag- and attribute names. In addition, each XML language is mapped out by a schema which encodes the syntactical structure, the compulsory and optional tags, and the allowed attribute and tag values. The main advantage of XML over plain text-only file formats lies in the structured way of data storage making it an extendable language that can easily be adjusted over time. To lower the specification burden and to increase the accessibility and recognizability, our XML standards closely follow the structure of the three-field notation explained in Section 3.

5.2. XML file templates and instance data repository

A first standard is used to store an instance of the problem and is made up of six different blocks of elements: meta data, data, resources, structure, constraints, and objective. Figure 5 shows a snippet of a fictive sports timetabling instance. To begin with, a meta data block stores some descriptive information such as the name of the instance, the name of the contributor, and the date of creation. Next, a data block holds information needed for the evaluation of a solution. Examples include pairwise venue distances, cost estimations, and weights for the weighted carryover or minimum break objective. Then, a resource block defines all team groups, teams, time groups, and time slots. We hereby note that time and team groups can be defined at the users own convenience to simplify the constraint formulation. The last three blocks encode the three-field notation as outlined in Section 3. First, the structure block encodes the α-field. For each team group corresponding to a league, a tag can be defined that describes the tournament format, the compactness of that tournament, and the symmetry structure. For non-round-robin formats, a games tag allows the user to enumerate additional games individually. Next, the constraints block enumerates all the constraints that are present in the instance. To enhance the compactness of the format, ‘teamGroup’ and ‘timeGroup’ attributes allow for comma-delimited lists of respectively team and time groups. The last tag encodes the objective in use.

A second standard is used to store solutions: a meta data tag first stores information such as the name of the corresponding problem instance, after which all scheduled games are enumerated. Finally, a third standard stores lower bounds (all problems are minimization problems; multiply costs with minus one for the revenue maximization objective). The first tag in the bound file stores meta data similarly to the solution file; the second tag contains the actual lower bounds on the infeasibility and objective value. A fragment of a solution and bound file is shown in Figure 6.

It is hard to assess the algorithmic performance of the various solution methods proposed in the literature for two reasons. First, since no set of benchmark problem instances is publicly available, many contributions in the literature propose a solution method which is tested on just one or two specific problem instances of a single competition. Second, the lack of generally accepted open challenge problems make that algorithmic progress over time is not monitored. For these reasons, little algorithmic understanding is gained from previous studies. To promote proper benchmarking, we offer a problem instance data repository containing various XML files encoding the majority of the problem instances published over the last five decades. In total, the repository contains artificial and real-world instances from over 15 different countries and eight different sports. For a full description of these files, we refer to Van Bulck et al. (2018a). The repository is publicly available (see Van Bulck et al. (2018b)) and will be continuously updated as new instances or better solutions become available.

5.3. File generation, validation tools, and online platform

A potential drawback of the use of XML is that users may not be familiar with markup languages. To mitigate this drawback, we developed an online document editor that allows users to create the XML files without directly editing any markup (see

```xml
<Solution>
  <MetaData>
     <InstanceName>Utopia</InstanceName>
     <SolutionName>UtupiaSol</SolutionName>
     <ObjectiveValue infeasibility="0" objective="2"/>
  </MetaData>
  <Games>
     <Game home="1" away="2" slot="1">
     ...
  </Games>
</Solution>
```

```xml
<Bound>
   <MetaData>
     <InstanceName>Utopia</InstanceName>
     <BoundName>LPRelaxation</BoundName>
     <Contributor>John Doe</Contributor>
   </MetaData>
   <LowerBound>
      <Infeasibility>0</Infeasibility>
      <Objective>2</Objective>
   </LowerBound>
</Bound>
```

Figure 6: Example of the solution XML format (left), and lower bound XML format (right).

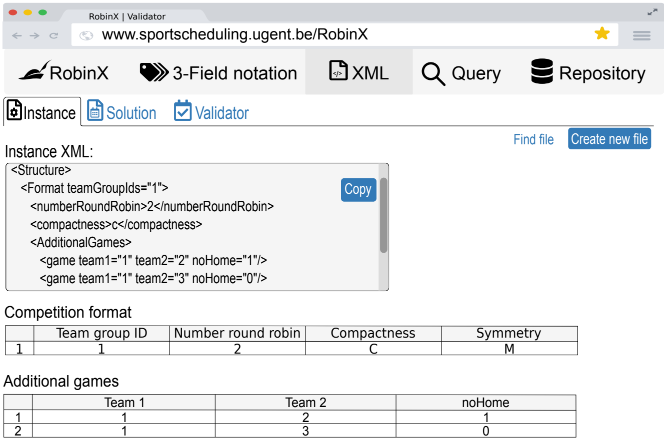

Figure 7: Illustration of the online web platform, step 1. Users can select XML files from the database or create new files.

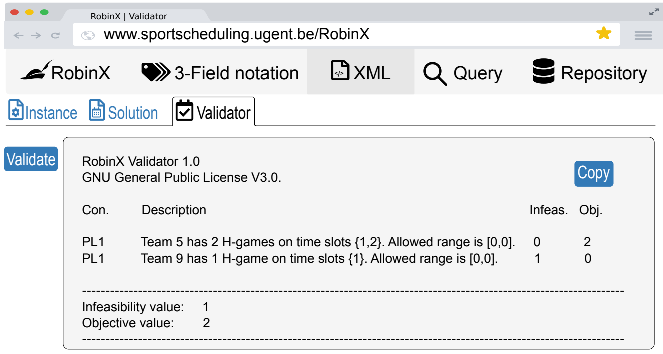

Figure 8: Illustration of the online web platform, step 2. To validate files, users only need to press the validate button.

Figure 7). Furthermore, we developed an open-source C++-library that is freely available for download (Van Bulck et al., 2018b). This library embeds a number of parsers that were used to encode the problem instances from the instance data repository of Section 5.2.

In addition, the library automates the validation of the files by performing syntactical checks on two different levels. First, an XML Schema Definition document checks whether the structure of the document is valid, and whether all required tags are specified. Next, the C++-library performs more advanced checks such as the verification whether referenced resources also have a corresponding resource tag. What is more, the library can also evaluate solution files. If a solution respects all hard constraints as described by the instance file, the program returns the objective value of the solution and all violated soft constraints. Otherwise, the program additionally returns the infeasibility value and a list with all violated soft and hard constraints. To facilitate the development of metaheuristics, the software also allows to validate solutions in an incremental fashion: if a timetable slightly changes, the program will only reevaluate the constraints affected by the change. To make the software more accessible, the library is embedded in a user-friendly web application. To validate solutions online, users simply select or create an instance and solution file and subsequently press the validation button. The program returns the infeasibility and objective value, and a list of all violated constraints (see Figure 8).

6. Conclusion

Over the past decades, the importance of finding fair, profitable, thrilling, or simply convenient timetables for sports leagues has introduced a wide variety of new constraints and objectives. This resulted in a proliferation of problem variants making it challenging for researchers to recognize the main structure of problem instances. This paper therefore proposed a three-field notation to describe (round-robin) sports timetabling problems. We used this notation to provide a problem-driven classification of real-life sports timetabling applications presented in the operations research literature. Moreover we presented a framework, called RobinX, to describe and exchange sports timetabling instances and solutions. RobinX can store nearly every constraint found in the literature and is complemented with a C++-library to validate and evaluate instances and solutions. In addition, we presented a problem instance data repository, which is unprecedented in size, containing over 40 different instance types. We hope that the automatic validation and exchange of problem instances and solutions will eventually lead to more general algorithmic insights. As an example, for which type of problem instances do integer programming methods work better than constraint programming? What characterizes instances on which an algorithm or decomposition method, e.g., first-break-then-schedule (Nemhauser and Trick, 1998) or first-schedule-then-break (Trick, 2001), performs poorly? What properties make an instance hard to solve? The structured data format of the XML files may assist future research to answer these questions via meta-learning techniques such as algorithm selection (see e.g., Smith-Miles and Lopes (2011)) or hyper-heuristics (see e.g., Burke et al. (2013)). Similar data sharing initiatives have revolutionized algorithmic development in other research disciplines such as high school timetabling (Post et al., 2012, 2014), nurse rostering (Kingston et al., 2018), multi-dimensional packing (Fekete and van der Veen, 2007), and supply chain modeling (Chatfield et al., 2009). With this paper, we invite researchers to join the project and submit their own problem instances and solutions.

References

Ball, B. C., Webster, D. B., 1977. Optimal scheduling for even-numbered team athletic conferences. Aiie T. 9, 161–169.

Bao, R., Trick, M., 2010. The relaxed traveling tournament problem. In: McCollum, B., Burke, E., White, G. (Eds.), Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling. PATAT, Belfast, pp. 472–476.

Bartsch, T., Drexl, A., Kröger, S., 2006. Scheduling the professional soccer leagues of Austria and Germany. Comput. Oper. Res. 33, 1907–1937.

Bean, J. C., Birge, J. R., 1980. Reducing travelling costs and player fatigue in the National Basketball Association. Interfaces 10, 98–102.

Bonomo, F., Cardemil, A., Durán, G., Marenco, J., Sabán, D., 2012. An application of the traveling tournament problem: The Argentine volleyball league. Interfaces 42 (3), 245–259.

Bradley, G., 2003. Introduction to extensible markup language (XML) with operations research examples. Newsletter INFORMS Computing Society 24 (1), 8–20.

Briskorn, D., 2009. Combinatorial properties of strength groups in round robin tournaments. EJOR 192, 744 – 754.

Briskorn, D., Drexl, A., 2009a. A branching scheme for finding cost-minimal round robin tournaments. EJOR 197, 68–76.

Briskorn, D., Drexl, A., 2009b. IP models for round robin tournaments. Comput. Oper. Res. 36, 837 – 852.

Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R., 2013. Hyper-heuristics: A survey of the state of the art. Journal of the Operational Research Society 64, 1695–1724.

Burrows, W., Tuffley, C. P., 2015. Maximising common fixtures in a round robin tournament with two divisions. Australasian Journal of Combinatorics 63, 153–169.

Cardoen, B., Demeulemeester, E., Beliën, J., 2010. Operating room planning and scheduling: A classification scheme. International Journal of Health Management and Information 1, 71–83.

Carlsson, M., Johansson, M., Larson, J., 2017. Scheduling double round-robin tournaments with divisional play using constraint programming. EJOR 259, 1180 – 1190.

Chatfield, D. C., Harrison, T. P., Hayya, J. C., 2009. SCML: An information framework to support supply chain modeling. EJOR 196, 651 – 660.

Cocchi, G., Galligari, A., Nicolino, F. P., Piccialli, V., Schoen, F., Sciandrone, M., 2018. Scheduling the Italian national volleyball tournament. Interfaces 48, 271–284.

Costa, D., 1994. An evolutionary tabu search algorithm and the NHL scheduling problem. Infor 33, 161–178.

de Werra, D., 1981. Scheduling in sports. In: Hansen, P. (Ed.), Studies on graphs and discrete programming. North-Holland, Amsterdam, pp. 381–395.

de Werra, D., 1985. On the multiplication of divisions: The use of graphs for sports scheduling. Networks 15, 125–136.

de Werra, D., Jacot-Descombes, L., Masson, P., 1990. A constrained sports scheduling problem. Discrete Appl. Math. 26, 41 – 49.

Della Croce, D., Oliveri, D., 2006. Scheduling the Italian Football League: an ILP-based approach. Comput. Oper. Res. 33, 1963 – 1974, special Issue: Operations Research in Sport.

Della Croce, F., Tadei, R., Asioli, P. S., 1999. Scheduling a round robin tennis tournament under courts and players availability constraints. Ann. Oper. Res. 92, 349–361.

Dinitz, J. H., Froncek, D., 2000. Scheduling the XFL. Congressus Numerantium, 5–17.

Drexl, A., Knust, S., 2007. Sports league scheduling: Graph- and resource-based models. Omega 35, 465 – 471.

Durán, G., Guajardo, M., Miranda, J., Sauré, D., Souyris, S., Weintraub, A., Wolf, R., 2007. Scheduling the Chilean soccer league by integer programming. Interfaces 37, 539–552.

Durán, G., Guajardo, M., Sauré, D., 2017. Scheduling the South American qualifiers to the 2018 FIFA World Cup by integer programming. EJOR 262, 1109 – 1115.

Durán, G., Guajardo, M., Wolf-Yadlin, R., 2012. Operations research techniques for scheduling Chile's second division soccer league. Interfaces 42, 273–285.

Easton, K., 2003. Using integer programming and constraint programming to solve sports scheduling problems. Ph.D. thesis, Georgia Institute of Technology, USA.

Easton, K., Nemhauser, G., Trick, M., 2001. The traveling tournament problem description and benchmarks. In: Walsh, T. (Ed.), Principles and Practice of Constraint Programming — CP 2001. Springer, Berlin, Heidelberg, pp. 580–584.

Fekete, S. P., van der Veen, J. C., 2007. PackLib2: An integrated library of multi-dimensional packing problems. EJOR 183, 1131 – 1135.

Ferland, J. A., Fleurent, C., 1991. Computer aided scheduling for a sport league. Infor 29, 14–25.

Fiallos, J., Pérez, J., Sabillón, F., Licona, M., 2010. Scheduling soccer league of Honduras using integer programming. In: IIE Annual Conference. Proceedings. Institute of Industrial and Systems Engineers (IISE), Institute of Industrial Engineers, p. 1.

Forrest, D., Simmons, R., 2006. New issues in attendance demand: The case of the English football league. J. Sports Econ. 7, 247–266.

Frončck, D., 2001. Scheduling the Czech national basketball league. Congressus Numerantium 153, 5–24.

Frončck, D., Meszka, M., 2005. Round robin tournaments with one bye and no breaks in home-away patterns are unique. In: Kendall, G., Burke, E. K., Petrovic, S., Gendreau, M. (Eds.), Multidisciplinary Scheduling: Theory and Applications. Springer, Boston, MA, pp. 331–340.

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

Goossens, D., Spieksma, F. C., 2012. The carryover effect does not influence football results. J. Sport. Econ. 13, 288–305.

Goossens, D. R., Spieksma, F. C., 2011. Soccer schedules in Europe: an overview. J. Sched. 15, 641–651.

Grabau, M., 2012. Softball scheduling as easy as 1-2-3 (strikes you're out). Interfaces 42, 310–319.

Graham, R. L., Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of discrete mathematics 5, 287–326.

Guedes, A. C. B., Ribeiro, C. C., 2011. A heuristic for minimizing weighted carry-over effects in round robin tournaments. J. Sched. 14, 655–667.

Günneç, D., Demir, E., 2018. Fair-fixture: Minimizing carry-over effects in football leagues. Journal of Industrial & Management Optimization 13, 1–16.

Hausken, M. D., Andersson, H., Fagerholt, K., Flatberg, T., 2012. Retracted: Scheduling the Norwegian football league. Int. T. Oper. Res. 20, 59–77.

Hoshino, K., Kawarabayashi, K., 2011a. A multi-round generalization of the traveling tournament problem and its application to Japanese baseball. EJOR 215, 481 – 497.

Hoshino, R., Kawarabayashi, K., 2011b. Scheduling bipartite tournaments to minimize total travel distance. J. Artif. Intell. Res. 42, 91–124.

Kendall, G., 2008. Scheduling English football fixtures over holiday periods. J. Oper. Res. Soc. 59, 743–755.

Kendall, G., Knust, S., Ribeiro, C. C., Urrutia, S., 2010. Scheduling in sports: An annotated bibliography. Computers & Operations Research 37, 1–19.

Kingston, J. H., 2001. Modelling timetabling problems with STTL. In: Burke, E., Erben, W. (Eds.), Practice and Theory of Automated Timetabling III. Springer Berlin, pp. 309–321.

Kingston, J. H., 2018. Specifying timetabling problems: an ongoing story. Invited Tutorial.

Kingston, J. H., Post, G., Vanden Berghe, G., 2018. A unified nurse rostering model based on XHSTT. In: Burke, E. K., Di Gaspero, L., McCollum, B., Musliu, N., Özcan, E. (Eds.), Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling. PATAT, Vienna, pp. 81–96.

Knust, S., 2010. Scheduling non-professional table-tennis leagues. EJOR 200, 358–367.

Knust, S., 2018. Classification of literature on sports scheduling. `http://www2.inf.uos.de/knust/sportssched/sportlit_class/`, accessed: 04/12/2018.

Knust, S., von Thaden, M., 2006. Balanced home–away assignments. Discrete Optim. 3, 354–365.

Kostuk, K. J., Willoughby, K. A., 2012. A decision support system for scheduling the Canadian Football League. Interfaces 42, 286–295.

Kyngäs, J., Nurmi, K., 2009a. Scheduling the Finnish 1st division ice hockey league. In: Proceedings of the twenty-second Florida Artificial Intelligence Research Society Conference. AAAI Press, Florida, pp. 195–200.

Kyngäs, J., Nurmi, K., 2009b. Scheduling the Finnish Major Ice Hockey League. In: Computational Intelligence in Scheduling, 2009. CI-Sched'09. IEEE Symposium on. IEEE, pp. 84–89.

Kyngäs, J., Nurmi, K., Kyngäs, N., Lilley, G., Salter, T., Goossens, D., 2017. Scheduling the Australian Football League. J. Oper. Res. Soc. 68, 973–982.

Larson, J., Johansson, M., 2014. Constructing schedules for sports leagues with divisional and round-robin tournaments. Journal of Quantitative Analysis in Sports 10, 119–129.

Lewis, R., Thompson, J., 2011. On the application of graph colouring techniques in round-robin sports scheduling. Comput. Oper. Res. 38, 190 – 204, project Management and Scheduling.

McAloon, K., Tretkoff, C., Wetzel, G., 1997. Sports league scheduling. In: Proceedings of Third ILOG opitmization suite International Users' Conference. Citeseer, Paris, France, pp. 1–5.

Nemhauser, G. L., Trick, M. A., 1998. Scheduling a major college basketball conference. Oper. Res. 46, 1–8.

Nurmi, K., Goossens, D., Bartsch, T., Bonomo, F., Briskorn, D., Durań, G., Kyngäs, J., Marenco, J., Ribeiro, C. C., Spieksma, F., Urrutia, S., Wolf, R., 2010. A framework for scheduling professional sports leagues. In: Katagir, H., Xu, L., Chan, A. H. (Eds.), Ao,S-I. Vol. 5. Springer, p. 1428.

Nurmi, K., Goossens, D., Kyngäs, J., 2013. Scheduling a triple round robin tournament with minitournaments for the Finnish national youth ice hockey league. J. Oper. Res. Soc. 65, 1770–1779.

Nurmi, K., Kyngäs, J., Goossens, D., Kyngäs, N., 2015. Scheduling the Finnish Major Ice Hockey League using the PEAST algorithm. In: Transactions on Engineering Technologies. Springer, pp. 155–168.

Pollard, R., Pollard, G., 2005. Long-term trends in home advantage in professional team sports in North America and England (1876–2003). J. Sport. Sci. 23 (4), 337–350.

Post, G., Ahmadi, S., Daskalaki, S., Kingston, J. H., Kyngäs, J., Nurmi, C., Ranson, D., 2012. An XML format for benchmarks in high school timetabling. Ann. Oper. Res. 194, 385–397.

Post, G., Kingston, J. H., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngäs, J., Nurmi, C., Musliu, N., Pillay, N., Santos, H., et al., 2014. XHSTT: an XML archive for high school timetabling problems in different countries. Ann. Oper. Res. 218, 295–301.

Rasmussen, R. V., 2008. Scheduling a triple round robin tournament for the best Danish soccer league. EJOR 185, 795 – 810.

Rasmussen, R. V., Trick, M. A., 2007. A benders approach for the constrained minimum break problem. EJOR 177, 198 – 213.

Rasmussen, R. V., Trick, M. A., 2008. Round robin scheduling–a survey. EJOR 188, 617–636.

Recalde, D., Torres, R., Vaca, P., 2013. Scheduling the professional Ecuadorian football league by integer programming. Comput. & Oper. Res. 40, 2478 – 2484.

Ribeiro, C. C., 2012. Sports scheduling: Problems and applications. Int. T. Oper. Res. 19, 201–226.

Ribeiro, C. C., Urrutia, S., 2007a. Heuristics for the mirrored traveling tournament problem. EJOR 179, 775–787.

Ribeiro, C. C., Urrutia, S., 2007b. Scheduling the Brazilian soccer tournament with fairness and broadcast objectives. In: Burke, E. K., Rudová, H. (Eds.), Practice and Theory of Automated Timetabling VI. Springer, Berlin, Heidelberg, pp. 147–157.

Ribeiro, C. C., Urrutia, S., 2012. Scheduling the Brazilian soccer tournament: Solution approach and practice. INFORMS Journal on Applied Analytics 42, 260–272.

Russell, K. G., 1980. Balancing carry-over effects in round robin tournaments. Biometrika 67, 127–131.

Russell, R. A., Leung, J. M. Y., 1994. Devising a cost effective schedule for a baseball league. Oper. Res. 42, 614–625.

Saur, M. C., Starr, K., Husted, M., Newman, A. M., 2012. Scheduling softball series in the Rocky Mountain Athletic Conference. Interfaces 42, 296–309.

Schauz, U., 2016. The tournament scheduling problem with absences. EJOR 254, 746 – 754.

Schönberger, J., 2017. The championship timetabling problem-construction and justification of test cases. In: Proceedings of MathSport International 2017 Conference. MathSport, p. 330.

Schönberger, J., Mattfeld, D. C., Kopfer, H., 2000. Automated timetable generation for rounds of a table-tennis league. In: Evolutionary Computation, 2000. Proceedings of the 2000 Congress on. Vol. 1. IEEE, pp. 277–284.

Schönberger, J., Mattfeld, D. C., Kopfer, H., 2004. Memetic algorithm timetabling for non-commercial sport leagues. EJOR 153, 102–116.

Schreuder, J. A. M., 1992. Combinatorial aspects of construction of competition Dutch professional football leagues. Discrete Appl. Math. 35, 301–312.

Smith-Miles, K., Lopes, L., 2011. Generalising algorithm performance in instance space: A timetabling case study. In: Coello, C. A. C. (Ed.), Learning and Intelligent Optimization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 524–538.

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

Trick, M. A., 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.

Urban, T., Russell, R. A., 2003. Scheduling sports competitions on multiple venues. EJOR 148 (2), 302 – 311.

Urrutia, S., Ribeiro, C. C., 2006. Maximizing breaks and bounding solutions to the mirrored traveling tournament problem. Discrete Appl. Math. 154, 1932–1938.

Van Bulck, D., Goossens, D., Schönberger, J., Guajardo, M., 2018a. An instance data repository for the round-robin sports timetabling problem. In: Submitted to proceedings of the first Conference of Mathsport Asia, under revision. pp. 1–13.

Van Bulck, D., Goossens, D., Schönberger, J., Guajardo, M., 2018b. RobinX: an XML-driven classification for round-robin sports timetabling – website. `http://www.sportscheduling.ugent.be/RobinX/`.

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

Van Voorhis, T., 2005. College basketball scheduling with travel swings. Comput. Ind. Eng. 48, 163 – 172.

Westphal, S., 2014. Scheduling the German basketball league. Interfaces 44, 498–508.

Wright, M., 1994. Timetabling county cricket fixtures using a form of tabu search. J. Oper. Res. Soc. 45, 758–770.

Wright, M., 2006. Scheduling fixtures for basketball New Zealand. Comput. & Oper. Res. 33, 1875 – 1893, special Issue: Operations Research in Sport.

Wright, M. B., 2005. Scheduling fixtures for New Zealand Cricket. Ima J. Manag. Math. 16, 99–112.

Zeng, L., Mizuno, S., 2012. On the separation in 2-period double round robin tournaments with minimum breaks. Comput. Oper. Res. 39 (7), 1692 – 1700.

Zhang, H., 2002. Generating college conference basketball schedules by a SAT solver. In: Proceedings of the fifth international symposium on the theory and applications of satisfiability testing. pp. 281–291.