Round robin scheduling - a survey

Rasmus V. Rasmussen and Michael A. Trick

Department of Operations Research, University of Aarhus, Ny Munkegade, Building 1530, 8000 Aarhus C, Denmark
Tepper School of Business, Carnegie Mellon University, Pittsburgh PA 15213, USA

Abstract This paper presents a comprehensive survey on the literature considering round robin tournaments. The terminology used within the area has been modified over time and today it is highly inconsistent. By presenting a coherent explanation of the various notions we hope that this paper will help to obtain a unified terminology. Furthermore, we outline the contributions presented during the last 30 years. The papers are divided into two categories (papers focusing on break minimization and papers focusing on distance minimization) and within each category we discuss the development which has taken place. Finally, we conclude the paper by discussing directions for future research within the area.
Keywords: Timetabling; Sports scheduling; round robin tournaments; home-away patterns.

1 Introduction

As long as there has been competitive sport, there has been a need for sports schedules. During the last 30 years, sports scheduling has turned into a research area of its own within the operations research and computer science communities. While it may seem trivial to schedule a tournament, and combinatorial mathematics has methods for scheduling simple tournaments, when additional requirements are added the problem becomes a very hard combinatorial optimization problem. In fact, for many types of problems, instances with more than 20 teams are considered large-scale and heuristic solution methods are often necessary in order to find good schedules.

The challenging problems and the practical applications provide a perfect area for developing and testing solution methods. In the literature we find methods ranging from pure combinatorial approaches to every aspect of discrete optimization, including integer programming (IP), constraint programming (CP), metaheuristic approaches, and various combinations thereof. The solution methods have evolved over time and today methods exist capable of finding optimal or near-optimal solutions for hard practical instances.

In addition to the theoretical gains from developing efficient solution methods capable of solving practical applications, sports scheduling has an economic aspect. Professional sports are big business and the revenue of a sports league may be affected by the quality of the schedule since a substantial part of the revenue often comes from TV networks. The TV networks buy the rights to broadcast the games but in return they want the most attractive games to be scheduled at certain dates.

In this paper we give a comprehensive survey of the sports scheduling literature concerned with scheduling round robin tournaments. The literature is partitioned into papers on break

minimization and papers on distance minimization. For both parts we present the main contributions and outline the development which has taken place. In order to keep the paper within reasonable size we have restricted ourselves to papers on round robin tournament problems in which the definition of the home or away status is relevant. This means that the problem of finding balanced tournament designs is not considered but for readers interested in this subject we refer to [7, 8, 20, 21, 29, 30, 45, 52, 60].

The rest of the paper is organized as follows. In Section 2, we present the terminology used within sports scheduling and, in Section 3, the various constraint types are outlined. The literature on break minimization and distance minimization are discussed in Section 4 and Section 5, respectively. Finally, Section 6 gives some concluding remarks and points out directions for future research.

2 Terminology

In this section, we define the sports scheduling terminology. It is important to stress that the terminology is far from consistent in the literature and some commonly used phrases have multiple meanings. However, to avoid misunderstandings, we will use the definitions from this section throughout the paper although it may conflict with papers to which we refer.

A *round robin tournament* is a tournament where all teams meet all other teams a fixed number of times. Most sports leagues play a double round robin tournament where teams meet twice but single, triple and quadruple round robin tournaments do also occur.

When scheduling a tournament, the games must be allocated to a number of *time slots* (slots) in such a way that each team plays at most one game in each slot. When the number of teams n is even at least $(n-1)$ slots are required and when n is odd at least n slots are required to schedule a single round robin tournament. In the case the number of available slots equals the lower bound, we say that the tournament is *compact* while it is *relaxed* when more slots are available. Note that these terms correspond to the terms *temporally constrained* and *temporally relaxed* defined in [36].

The allocation of games to slots can be presented as a *timetable*. Each row of the timetable corresponds to a team while the columns correspond to slots. The entry of row i and column s is the opponent of team i in slot s. Figure 1 shows a timetable for a compact single round robin tournament with 6 teams and a timetable for a corresponding tournament with 7 teams.

Slots	1	2	3	4	5
Team 1	6	3	5	2	4
Team 2	5	6	4	1	3
Team 3	4	1	6	5	2
Team 4	3	5	2	6	1
Team 5	2	4	1	3	6
Team 6	1	2	3	4	5

Slots	1	2	3	4	5	6	7
Team 1	7		5	2	4	6	3
Team 2	5	6		1	3	7	4
Team 3	4	7	6		2	5	1
Team 4	3	5	7	6	1		2
Team 5	2	4	1	7	6	3	
Team 6		2	3	4	5	1	7
Team 7	1	3	4	5		2	6

Figure 1: Examples of timetables for tournaments with 6 and 7 teams.

In the literature, teams often have an associated *venue* and when they play at their own venue, they play *home games* while they play *away games* at all other venues. In slots without a game they are said to have a *bye*. It is assumed that each time two teams meet, one of the teams plays home while the other plays away. For single round-robin schedules, it is often required that

the deviation between the number of home games and away games played by each team is no more than 1. Such a schedule is termed *balanced*. For double round-robin schedules, it is typically required that the two games between every pair of opponents occur in opposite venues. Such a schedule is forced to be balanced. For multi round-robin schedules, a common requirement is that the schedule is both balanced and that for every pair of teams, the the deviation between venues of games for that pair be no more than 1.

The sequence of home games, away games and byes according to which a team plays during the tournament is known as a *home away pattern* (pattern). If byes occur in the tournament, a pattern is normally represented by a vector with an entry for each slot containing either an H, an A, or a B. In compact tournaments with an even number of teams, all teams play in each slot and the B is omitted. In this case H and A are often replaced by 1 and 0, respectively. In many tournaments it is considered attractive to have an alternating pattern of home and away games and a pattern is said to have a *break* in slots differing from such an alternating sequence. This means that a break corresponds to two consecutive home games or two consecutive away games. Two patterns are said to be *complementary* if the first pattern has an away game when the second pattern has a home game and vice versa. Figure 2 (a) shows 2 complementary patterns for a compact single round robin tournament with 6 teams. Notice that both patterns have a break in slot 3.

1	0	0	1	0
0	1	1	0	1

(a)

Slots	1	2	3	4	5
Team 1	1	0	1	0	1
Team 2	1	0	0	1	0
Team 3	0	1	1	0	1
Team 4	1	0	1	0	0
Team 5	0	1	0	1	1
Team 6	0	1	0	1	0

(b)

Figure 2: (a) Two complementary patterns, (b) Example of a pattern set for a tournament with 6 teams.

To represent the assignments of home and away games for a tournament with n teams, we use a *home away pattern set* (pattern set). This is a set of exactly n patterns and each pattern is associated with one of the teams. Figure 2 (b) shows an example of a pattern set for a tournament with 6 teams. Notice that this pattern set exclusively consists of pairs of complementary patterns. When this is the case, the pattern set satisfies the *complementary property* and it is said to be *complementary*. If all teams have the same number of breaks it is an *equitable* pattern set.

Furthermore, the pattern set can be associated with the timetable for 6 teams displayed in Figure 1 since, in every game, one of the opponents plays home while the other plays away. A pattern set for which a corresponding timetable exists is said to be *feasible*. Figure 3 gives an example of three patterns which would make a pattern set *infeasible* since the three mutual games can only be played in slots 1 and 2.

The combination of a pattern set and a corresponding timetable constitutes a *schedule* for a tournament. A schedule is *mirrored* when the first and the second half are identical except the home games and away games are exchanged. Furthermore, we say that a schedule for a single round robin tournament is *irreducible* when at most one opponent in each game has a break. A schedule can be represented as in Figure 4 showing a mirrored double round robin schedule. In the figure, a $+$ denotes a home game while a $-$ denotes an away game. A sequence of consecutive

1	0	0	1	0
1	1	0	1	0
0	1	0	1	0

Figure 3: Example of an infeasible subset of patterns.

away games is called a *trip* while a sequence of consecutive home games is called a *home stand*. An entire row of the schedule defines a *tour* for the corresponding team.

Slots	1	2	3	4	5	6	7	8	9	10
Team 1	+6	−3	+5	−2	+4	−6	+3	−5	+2	−4
Team 2	+5	−6	−4	+1	−3	−5	+6	+4	−1	+3
Team 3	−4	+1	+6	−5	+2	+4	−1	−6	+5	−2
Team 4	+3	−5	+2	−6	−1	−3	+5	−2	+6	+1
Team 5	−2	+4	−1	+3	+6	+2	−4	+1	−3	−6
Team 6	−1	+2	−3	+4	−5	+1	−2	+3	−4	+5

Figure 4: Example of a mirrored double round robin schedule.

When solving a sports scheduling problem it may be advantageous to postpone the assignment of games until a schedule has been obtained. In that case *placeholders* are used to represent the teams in the pattern set and in the timetable until a schedule has been found.

Since round robin tournaments have a correspondence to graphs, we also introduce a few graph theoretical concepts. Consider a graph $G = (V, E)$ where V is a finite set of nodes and E is the set of edges in G. A *matching* in G is a set of independent edges (non-adjacent edges) and a matching in which all the nodes in V are incident to an edge is called a *perfect matching*. The graph induced by a complete matching is a 1-regular graph (the degree is 1 for all nodes). This is called a 1-*factor* and a partitioning of the graph G into factors is called a *factorization*.

In the rest of the paper, we let n denote the number of teams, T the set of teams and S the set of slots. Since most of the sports scheduling literature focuses on compact tournaments with an even number of teams this is assumed to be the case unless otherwise stated.

3 Constraints

Practical sports scheduling applications are very often characterized by a large number of conflicting constraints arising from teams, TV networks, sports associations, fans and local communities. Consequently, a section discussing the various constraints applicable to a particular sports league has become a standard part of papers considering practical applications since each league has their own special requirements. In this section we will give a short outline of the most typical constraints and we give references to papers facing these constraints.

Place constraints ([4, 5, 9, 11–13, 17, 25, 38, 39, 41, 46, 48, 50, 54, 58, 62])
Constraints ensuring that a team plays home or away in a certain slot. This kind of constraint is normally imposed when a venue is unavailable due to other events.

Top team and bottom team constraints ([4, 12, 13, 23, 25, 36, 38, 46, 48])
In some leagues special considerations are taken for teams which have just qualified for the league and teams which are known to be strong.

Break constraints ([4, 13, 23, 25, 36, 38, 59])
Often leagues want to avoid a schedule where teams have a break in slot 2 or a break in the last slot.

Game constraints ([4, 9, 13, 16, 23, 26, 33–38, 49, 62])
These are constraints fixing a certain game to a particular time slot. The constraints are normally imposed by TV networks who want "big" games at certain dates.

Complementary constraints ([4, 9, 12, 13, 38, 46, 59])
When two teams share a venue, a complementary constraint is used to make sure that the two teams play home at different slots. Of course both teams play home in some sense when they meet but playing the official home game may be important since revenue is often earned by the home team.

Geographical constraints ([4, 12, 13, 38, 48, 55])
To avoid slots in which many home games are gathered in a small area games should be scattered throughout the region in which the tournament is played.

Pattern constraints ([2, 4, 5, 9–13, 17, 23, 25, 33, 36, 38, 41, 44, 46, 50, 54, 62])
Some applications have special requirements on the patterns such as restrictions on the number of consecutive breaks or certain sequences of home games, away games and byes which should be avoided. They also include requests for equitable pattern sets saying that all teams must have the same number of breaks.

Separation constraints ([4, 11, 17, 23, 36, 38, 39, 62])
When we consider tournaments where teams meet more than once and the schedule is non-mirrored, most leagues have a lower bound on the number of slots between two games with the same opponents. This constraint is not relevant to mirrored schedules since such schedules always have at least $n-2$ slots between such games.

In many applications the constraints are separated into hard constraints and soft constraints. All the hard constraints must be satisfied in a feasible solution, while the soft constraints are penalized such that penalties are incurred if the constraints are violated. In addition to minimizing the number of violated soft constraints the objective of a sports scheduling problem is normally to minimize either the number of breaks or the travel distance. In the following two sections we will discuss the papers on minimizing breaks and minimizing travel distance, respectively. For an online overview and classification of the literature we refer to the website `http://www.informatik.uni-osnabrueck.de/knust/sportlit_class/`.

4 1-Factorizations and Minimizing Breaks

A rich line of research has been to exploit the close relationship between 1-factorizations of graphs with tournaments. When venues are added, this leads to oriented 1-factorizations.

When teams return home after each away game instead of travelling from one away game to the next, alternating patterns of home and away games are usually preferred. Such patterns consider the fans by avoiding long periods without home games and they ensure regular earnings from home games.

The need for alternating patterns has led to a large amount of research originating from graph theoretical approaches for minimizing the number of breaks in a pattern set and leading to highly sophisticated solution methods for practical applications facing numerous constraints. During the last 30 years, focus has moved from constructive methods applicable for general tournaments without additional constraints to decomposition methods capable of handling all the constraints applicable for a certain sports league.

4.1 Constructive Methods

In the 1980's, Rosa and Wallis [43], de Werra [55, 56, 57, 58], de Werra et al. [59] and Schreuder [47] published a number of papers on the relationship between graphs and tournaments and used the relationship to obtain results for schedules. De Werra [56] presents the relationship between a tournament and a graph in the following way.

Consider a compact single round robin tournament with an even number of teams n - in case the number of teams is uneven a dummy team can be added. This tournament can be associated with the complete graph K_n by letting each node correspond to a team and letting each edge correspond to the game between the teams associated with the end nodes. A 1-factorization $F = (F_1, \ldots, F_{n-1})$ of K_n where $F_1, \ldots, F_{n-1}$ are 1-factors then corresponds to a partitioning of the games into $n-1$ slots since each node will be incident to exactly one edge in each 1-factor. We refer to Mendelsohn and Rosa [31] for a survey on 1-factorizations.

One direct approach to creating a timetable is to form it slot-by-slot. Rosa and Wallis [43] show that such an approach may fail. They define a *premature set* to be a partial timetable (only the first k slots are determined) which cannot be extended to a full timetable and ask the question: How much can go wrong if we assign games one slot at a time without looking ahead? In other words do premature sets exist? Indeed, they do exist and Rosa and Wallis prove the following corollary.

Corollary 1 (Rosa and Wallis [43]) *For n even, there is a premature set of k one-factors in K_n whenever $\frac{n}{2} \leq k \leq n-3$ and $\frac{n}{2}$ is odd, and whenever $\frac{n}{2} < k \leq n-3$ and $\frac{n}{2}$ is even.*

They also show that when the tournament is big enough nothing can go wrong in the first slots.

Corollary 2 (Rosa and Wallis [43]) *If $n \geq 8$ and even, there exists no premature set of three 1-factors in K_n.*

This corollary is followed by a conjecture which is still an open question.

Conjecture 1 (Rosa and Wallis [43]) *For any positive integer k, there exists $n(k)$ such that if $n > n(k)$, then any premature set of 1-factors of K_n contains more than k one-factors.*

The existence of premature sets precludes the possibility of a "greedy" or slot-by-slot approach to finding tournaments: such approaches can lead to a premature set.

The home away assignments can be represented by orienting the edges and letting an edge from node i to node j correspond to a game where team i visits team j. An oriented 1-factorization $\vec{F} = (\vec{F}_1, \ldots, \vec{F}_{n-1})$ or equivalently an oriented $(n-1)$-coloring then characterizes a schedule for the single round robin tournament. Figure 5 shows an oriented 1-factorization of K_6 and Figure 6 shows the associated schedule.

We present some of the most important results obtained from the relationship between graphs and schedules. The first and most basic result is the following.

Proposition 1 (De Werra [56]) *In any oriented coloring of K_n, there are at least $n-2$ breaks.*

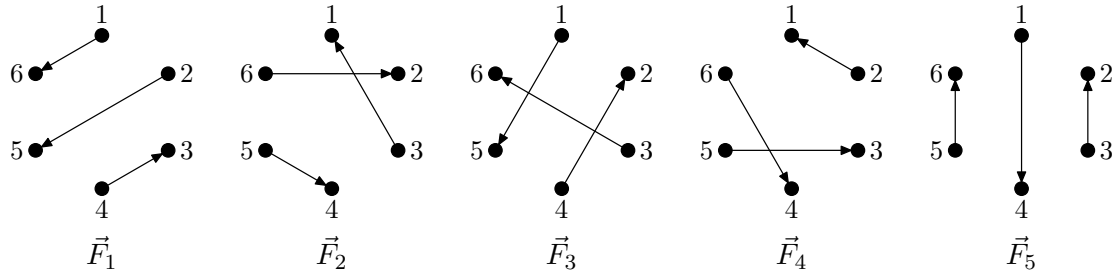


Figure 5: Oriented 1-factorization of K_6.

Slots	1	2	3	4	5
Team 1	-6	$+3$	-5	$+2$	-4
Team 2	-5	$+6$	$+4$	-1	$+3$
Team 3	$+4$	-1	-6	$+5$	-2
Team 4	-3	$+5$	-2	$+6$	$+1$
Team 5	$+2$	-4	$+1$	-3	-6
Team 6	$+1$	-2	$+3$	-4	$+5$

Figure 6: Schedule corresponding to the 1-factorization from Figure 5.

The proof is straightforward when observing that at most two teams can have a pattern without breaks. However, the result is very important since it gives a lower bound on the number of breaks in a single round robin tournament. Furthermore, de Werra was also able to show that the lower bound was obtainable by constructing a 1-factorization with exactly $n-2$ breaks. The 1-factorization is called the *canonical 1-factorization* and it is defined as follows.

Definition 1 (De Werra [56]) *The canonical 1-factorization satisfies that for* $i = 1, \ldots, n-1$

$$F_i = \{(n, i)\} \cup \{(i+k, i-k) : k = 1, \ldots, n/2 - 1\}$$

where $i + k$ *and* $i - k$ *are expressed as one of the numbers* $1, \ldots, n-1$ *(mod* $(n-1)$*).*

To obtain a schedule with exactly $n-2$ breaks, the canonical factorization is oriented such that the edge (i, n) is oriented from i to n if i is odd and from n to i if i is even and the edge $(i+k, i-k)$ in F_i is oriented from $i+k$ to $i-k$ if k is odd and the other way if k is even.

Proposition 2 (De Werra [56]) *There exists an oriented coloring of* K_n *with exactly* $n-2$ *breaks.*

The canonical 1-factorization has subsequently been widely used in the literature and the associated schedule is referred to as the *canonical schedule*. The factorization and schedule shown in Figure 5 and Figure 6 are the canonical factorization and the canonical schedule for a tournament with 6 teams.

The canonical schedule can also be used for tournaments with an uneven number of teams by using a dummy node and in this way de Werra was able to construct a tournament without breaks.

Corollary 3 (De Werra [56]) K_{n+1} *has an oriented coloring without breaks.*

Notice, that the removal of team 6 in Figure 6 produces a schedule for 5 teams without breaks.

Multi-period schedules were also considered and the following two results were obtained.

Proposition 3 (De Werra [56]) *A mirrored double round robin tournament has at least* $3n-6$ *breaks.*

This can be proved by noting that teams with 1 break in the first half have a corresponding break in the second half and a third break at the beginning of the second half.

Proposition 4 (De Werra [56]) *A mirrored double round robin tournament with exactly* $3n-6$ *breaks exists and if* $n \neq 4$ *no team has two consecutive breaks.*

Again the canonical schedule was used for constructing a mirrored double round robin tournament with exactly $3n - 6$ breaks although small modifications were necessary to avoid consecutive breaks. The resulting schedule is known as the *modified canonical schedule*.

In [55], de Werra gives a characterization of *canonically feasible break sequences*. A break sequence is a sequence telling at which slots the breaks occur. A given sequence is said to be canonically feasible if there exists an oriented coloring derived from a canonical 1-factorization having breaks occurring in pairs for colors which can be associated with the break sequence. De Werra also considers tournaments facing geographical constraints which require that, when teams are located close to each other, they should have complementary patterns if possible. De Werra treats a number of specific problems occurring when geographical constraints are considered and presents constructive methods for obtaining schedules.

Schreuder [47] formulates necessary and sufficient conditions for a single round robin tournament by using 0-1 variables $x_{i_1 i_2 s}$ which is 1 if team i_1 plays home against team i_2 in slot s. The conditions look as follows:

$$\sum_{i_1 \in T} (x_{i_1 i_2 s} + x_{i_2 i_1 s}) = 1 \qquad \forall i_2 \in T, \ \forall s \in S$$

$$\sum_{s \in S} (x_{i_1 i_2 s} + x_{i_2 i_1 s}) = 1 \qquad \forall i_1, i_2 \in T, \ i_1 \neq i_2$$

These constraints can be used to formulate round robin scheduling problems using integer programming although more sophisticated methods are needed in order to solve problems of realistic size.

In [57] de Werra concentrates on irregularities in schedules. He notices that when no breaks occur between two time slots s_1 and s_2 the edges of the oriented graph K_n, corresponding to the games played in the slots $s_1, \ldots, s_2$, will form a regular bipartite graph. This property is used to obtain schedules minimizing the number of *irregular slots* (slots containing a break) and to distribute the irregular slots evenly.

De Werra summarizes most of the previous results in [58] where, for the first time, place constraints are considered. In order to solve the scheduling problem with place constraints, schedules with placeholders are generated and, for each schedule, teams are assigned to placeholders by constructing a factor in a bipartite graph. The bipartite graph contains a node for each team, a node for each placeholder, and an edge between a team i and a placeholder j if the pattern of placeholder j satisfies the place constraints of team i. If a factor can be constructed in the bipartite graph we have a feasible solution and otherwise we move on to the next schedule. This is the first step towards the decomposition methods presented in the following section.

However, before moving to the decomposition methods, let us mention de Werra et al. [59] facing a problem with 2 leagues A and B. League A plays a double round robin while league B plays a single round robin before it is partitioned into two leagues C and C' which both play

an additional single round robin. The partitioning of league B is not known in advance since it depends on the outcome of the games. The objective is to spread breaks evenly and minimize the total number of breaks. The problem is constrained by teams from different leagues using the same venue, and breaks in the last slot are not allowed. Since team specific requirements are not considered, it is possible to construct an optimal solution for the problem.

4.2 The Constrained Minimum Break Problem

In the beginning of the 1990's focus moved from the graph theoretical results to practical applications. This change meant that the constraints outlined in Section 3 were taken into account and solution methods capable of handling these constraints had to be developed. The problem of finding a schedule minimizing the number of breaks and at the same time take additional constraints into account is known as the *constrained minimum break problem*. However, the problem may change significantly from one application to another since different constraints are considered.

To solve the problem two metaheuristic approaches were applied by Willis and Terrill [61] who use simulated annealing and Wright [63] who uses tabu search for scheduling cricket tournaments. However, the majority of the papers use a decomposition approach. A sports scheduling problem naturally decomposes into four steps and, although the order of the steps vary and some steps are combined, these four steps are used in almost all solution methods for solving variations of the constrained minimum break problem. The four steps are:

Step 1 Generate patterns.

Step 2 Find a pattern set for placeholders.

Step 3 Find a timetable for placeholders.

Step 4 Allocate teams to placeholders.

Schreuder [48] solves a mirrored double round robin problem for the Dutch professional football league and uses a 2-phase approach which resembles the method used by de Werra [58]. In this method Phase 1 combines Steps 1 to 3 by constructing the canonical schedule for placeholders and Phase 2 corresponds to Step 4 and allocates teams to placeholders. The problem of assigning teams to placeholders is formulated as a quadratic assignment problem and a heuristic solution method is presented for solving the problem.

In 1998 Nemhauser and Trick [36] schedule the basketball tournament for the Atlantic Coast Conference consisting of nine university teams from the United States. In their approach all four steps are used but instead of using a combinatorial design, as seen in the earlier approaches, they use IP combined with enumeration techniques to obtain pattern sets. In Step 1 they generate mirrored patterns having a reasonable chance of being used in a feasible pattern set and in order to satisfy a specific constraint, slots 8 and 10 are interchanged. After the patterns have been generated, an IP model is used in Step 2 to generate pattern sets. The IP model chooses 9 patterns which minimize the number of breaks and it requires that in each slot, 4 patterns have a home game, 4 patterns have an away game and 1 pattern has a bye. All feasible solutions to the model are generated and it leads to 17 pattern sets. For each pattern set all feasible timetables are generated using another IP model and this leads to 826 timetables. Finally, teams are allocated to placeholders by enumerating through the 9! possible allocations. Almost 300 million schedules had to be considered but only 17 were feasible and from these schedules a final schedule was chosen.

After the IP/enumeration approach by Nemhauser and Trick, Schaerf [46], Henz [23, 25], and Régin [41] introduced CP approaches for solving sports scheduling problems.

Schaerf [46] considers the problem of scheduling a mirrored double round robin tournament with complementary constraints, place constraints, geographic constraints and top team constraints. The constraints are split into hard constraints which must be satisfied and soft constraints enforcing a penalty when violated. To solve the problem, he uses the 2-phase approach known from de Werra [58] and Schreuder [48] in which Phase 1 combines Steps 1, 2 and 3 while Phase 2 corresponds to Step 4. Phase 1 is handled by using the modified canonical schedule since this schedule minimizes the number of breaks and avoids consecutive breaks but it is noted that Phase 2 is independent of the schedule chosen in Phase 1. The assignment problem considered in Phase 2 is solved using CP. The variables and constraints used to formulate the problem are outlined and computational results are presented. The CP model takes longer time than the heuristic method presented by Schreuder [48] but in return it gives the optimal solution.

In contrast to Schaerf [46], Henz [25] uses CP to solve all four steps. The individual steps are solved in the order 1, 2, 3, 4 and in the order 1, 2, 4, 3. Henz reports that, in most cases, the best performances are obtained by solving Step 4 before Step 3. CP models are presented for each of the four steps and a generic constraint-based round robin planning tool known as Friar Tuck is presented. Friar Tuck uses the finite domain constraint programming system Mozart 1.0 and allows the user to fine-tune the solution process and the constraints. In [23] Henz uses the CP approach explained in [25] to solve the Basketball league considered by Nemhauser and Trick and shows that the CP approach clearly outperforms the combined IP and enumeration technique used previously. Henz is able to find all solutions to the problem in less than one minute while Nemhauser and Trick used more than 24 hours.

Régin [41] also presents CP approaches for solving sports scheduling problems. At first he gives a general discussion of symmetry breaking constraints, the use of implicit constraints, global constraints and pertinent and redundant constraints. This discussion is followed by a CP model for generating a single round robin schedule with a minimal number of breaks when no additional constraints are present. Notice, that the canonical schedule also solves this problem. Régin shows how symmetry breaking is able to enhance performance significantly and the problem size solvable in approximately 1 minute increases from 6 to 60 teams. Next Régin considers a problem which is later known as *the break minimization problem*.

Definition 2 *Given a timetable, the break minimization problem consists of finding a feasible pattern set which minimizes the number of breaks.*

For this problem most of the symmetry breaking constraints added to the first model become invalid but Régin is able to derive new constraints and again significant improvements can be obtained. In this case a problem with 16 teams can be solved in approximately 1 minute.

Subsequently, Trick [49] motivates the use of the break minimization problem by discussing the order of the four solution steps. He argues that the steps should be ordered such that the most critical aspects of the schedule are considered early in the solution process. Solving Steps 1 and 2 before Steps 3 and 4 makes sense when for instance many place constraints are considered. On the other hand, when game constraints or other constraints associated with the timetable become more important, Steps 3 and 4 should be solved before Steps 1 and 2. Trick presents a 2-phase solution method which solves Steps 3 and 4 in Phase 1 and solves the break minimization problem corresponding to Steps 1 and 2 in Phase 2. The method combines CP and IP by using CP for Phase 1 and IP for Phase 2. Two CP models for solving Phase 1 are discussed and both models are able to find a 20-team schedule in less than 1 second and able to find 500 20-team schedules in around one minute. In Phase 2 the symmetry breaking constraints presented by Régin [41] are used in an IP model and the computation times show improvements for large

instances (more than 16 teams) compared to the CP model presented by Régin.

The papers by Régin [41] and Trick [49] were followed by a number of papers focusing on the break minimization problem alone. Elf et al. [16] show that solving the break minimization problem is equivalent to a maximum cut problem in an undirected graph G. Given a timetable, the graph G is constructed by adding a node v_{is} for each team i and each slot s such that v_{is} corresponds to entry (i, s) in the timetable. An example is shown in Figure 7.

Slots	1	2	3	4	5
Team 1	6	3	5	2	4
Team 2	5	6	4	1	3
Team 3	4	1	6	5	2
Team 4	3	5	2	6	1
Team 5	2	4	1	3	6
Team 6	1	2	3	4	5

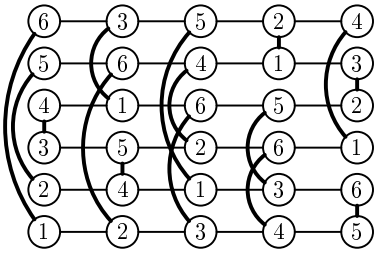


Figure 7: Timetable and corresponding maximum cut graph.

For each team i and each slot s in $2, \ldots, |S|$ the nodes v_{is-1} and v_{is} are connected by an edge corresponding to the horizontal edges in Figure 7. The vertical edges combine nodes $v_{i_1 s}$ and $v_{i_2 s}$ when team i_1 plays against team i_2 in slot s. By assigning a weight of 1 to all the horizontal edges and a weight M to the vertical edges, Elf et al. are able to show that a maximum cut in G corresponds to an optimal solution to the break minimization problem when $M \geq n(n-2)+1$. The reasoning behind the argument is that a cut separates the vertices into two sets. One of the sets will correspond to home games and the other to away games. In order to obtain a feasible home-away assignment we must ensure that, when two nodes play against each other, one belongs to the set of home games while the other belongs to the set of away games. This is handled by assigning big weights to all the vertical edges in G. Maximizing the number of horizontal edges in the cut, corresponds to minimizing the number of breaks since an edge which is not part of the cut leads to a break.

After the graph G has been constructed, Elf et al. show how to transform this graph into a smaller graph by contracting the vertical edges one by one and changing the signs of some of the horizontal edges. This leads to a graph with $\frac{n(n-1)}{2}$ nodes and $n(n-1)$ edges. The modified graph speeds up the solution process since a maximum cut for the modified graph can be directly transformed to a maximum cut for the original graph G. A maximum cut is found by applying a branch and cut algorithm described by Barahona, Grötschel, Jünger, and Reinelt [3]. The computational tests show great reductions in computation times compared to the CP and IP approaches presented by Régin and Trick, respectively, and instances with up to 26 teams can be solved within reasonable time (1215.9 seconds).

A similar idea is used by Miyashiro and Matsui [34] who also consider the break minimization problem. They use two graphs G_1 and G_2 both having a node set equal to the node set of G. The edges of G_1 correspond to the horizontal edges of G while the edges of G_2 correspond to the vertical edges of G. Instead of using weights equal to M, they notice that the problem is a special case of MAX RES CUT discussed by Goemans and Williamson [19] and therefore solvable by an approximation algorithm based on positive semidefinite programming proposed by Goemans and Williamson [19]. The problem can also be stated as a special case of MAX 2SAT but solving the MAX 2SAT problem is equivalent to solving the MAX RES CUT problem when the algorithm of Goemans and Williamson is applied. In contrast to the previous methods on the break minimization problem, this is an approximative method and this makes it capable

of finding solutions for problems with up to 40 teams compared to the 26 teams considered by Elf et al. [16].

At the end of the paper by Elf et al. [16], it is conjectured that instances with only $n-2$ breaks are solvable in polynomial time and that the break minimization problem in general is NP-hard. The second conjecture is still an open problem.

The first conjecture was proved affirmatively in [32] where Miyashiro and Matsui consider the problem of finding a pattern set with exactly $n-2$ breaks for a given timetable or showing that such a pattern set does not exist. The problem is reduced into n decision problems $\mathrm{P1}_k$ for $k = 1, \ldots, n$ where $\mathrm{P1}_k$ is similar to the original problem except for an extra constraint requiring that team k has a pattern without breaks and starts with a home game. If a pattern set exists for one of the problems $\mathrm{P1}_1, \ldots, \mathrm{P1}_n$, we have a solution and otherwise no feasible pattern set exists with $n-2$ breaks. Since 2SAT problems can be solved in polynomial time, Miyashiro and Matsui are now able to show that the original problem can be solved in polynomial time by transforming each of the problems $\mathrm{P1}_1, \ldots \mathrm{P1}_n$ into a 2SAT problem. The transformation is accomplished by constructing a pattern set and using boolean variables x_{is} which are true when team i plays according to the constructed pattern set in slot s and false otherwise. The conclusion is that, for a given timetable, it is possible to find a feasible pattern set with $n-2$ breaks in polynomial time or show that such a pattern set does not exist.

Corollary 4 (Miyashiro and Matsui [32]) *The following problem is solvable in polynomial time.*

Instance: *A timetable with n teams where n is even.*

Task: *Find a pattern set with at most $n-2$ breaks that is consistent with the given timetable if it exists and return "infeasible" otherwise.*

Furthermore, Miyashiro and Matsui [33] show that the result is also valid for pattern sets with n breaks.

Corollary 5 (Miyashiro and Matsui [33]) *The following problem is solvable in polynomial time.*

Instance: *A timetable with n teams where n is even.*

Task: *Find a pattern set with at most n breaks that is consistent with the given timetable if it exists and return "infeasible" otherwise.*

The procedure is very similar to the procedure used in [32]. Again the problem is transformed to a number of 2SAT problems and since they can be solved in polynomial time, it is possible to solve the original problem in polynomial time. In addition to the corollary an interesting property combining break minimization and break maximization is presented. Given a pattern set H represented by a h_{is} for each team i and each slot s, the pattern set $\tilde{H}$ is defined such that $\tilde{h}_{is} = h_{is}$ if s is uneven and $\tilde{h}_{is} \neq h_{is}$ if s is even. Due to the construction, each team has a break in each slot s, $s \geq 2$, in exactly one of the pattern sets and this leads to the following lemma.

Lemma 1 (Miyashiro and Matsui [33]) *Let H be a pattern set for a tournament with n teams where n is even. Then the number of breaks in H plus the number of breaks in $\tilde{H}$ equals $n(n-2)$.*

Lemma 1 implies the following theorem.

Theorem 1 (Miyashiro and Matsui [33]) *Given a timetable, then a feasible pattern set minimizes the number of breaks if and only if $\bar{H}$ maximizes the number of breaks.*

This implies that minimizing and maximizing the number of breaks for a given timetable is equivalent.

A variant of the second conjecture by Elf et al. [16] regarding NP-hardness of the break minimization problem is considered by Post and Woeginger [37]. They consider *partial timetables* for single round robin tournaments. The partial timetables only contain a subset of the normal $n-1$ slots and they satisfy that two teams do not meet more than once. Break minimization in a partial timetable means finding a home away assignment for the partial timetable such that the number of breaks is minimized. By using a polynomial time reduction from an NP-hard version of the Max-Cut problem Post and Woeginger are able to show the following theorem.

Theorem 2 (Post and Woeginger [37]) *Break minimization in partial timetables with n teams and three slots is NP-hard.*

The theorem leads to the following corollary.

Corollary 6 (Post and Woeginger [37]) *Break minimization in partial timetables with n teams and a fixed number $r \geq 4$ of slots is NP-hard.*

Post and Woeginger also consider lower and upper bounds on the solution values for the break minimization problem. Let $B_{min}(TT_n)$ be the optimal solution value to the break minimization problem given the timetable TT_n with n teams. They are able to obtain a lower bound on $\max_{TT_n} B_{min}(TT_n)$ when $n = 4^k$ for some $k \geq 1$.

Theorem 3 (Post and Woeginger [37]) *For $n = 4^k$ teams with $k \geq 1$, there exists a timetable TT_n^* with $B_{min}(TT_n^*) \geq \frac{1}{6}n(n-1)$.*

An upper bound on $B_{min}(TT_n)$ for an arbitrary timetable TT_n is also derived.

Theorem 4 (Post and Woeginger [37]) *Each timetable TT_n for n teams satisfies*

$$B_{min}(TT_n) \leq \begin{cases} \frac{1}{4}n(n-2), & \text{if } n \text{ is of the form } 4k; \\ \frac{1}{4}(n-2)^2, & \text{if } n \text{ is of the form } 4k+2. \end{cases}$$

Furthermore, a corresponding pattern set can be computed in polynomial time.

In Table 1 the lower bounds obtained by Elf et al. [16] are denoted LB-EJR, the lower bounds for schedules with $n = 4^k$ are denoted LB-PW and the upper bounds are stated UB-PW according to the table from [37].

Table 1: Upper and lower bounds for $\max_{TT_n} B_{min}(TT_n)$ with $n \leq 26$.

n	4	6	8	10	12	14	16	18	20	22	24	26
LB-EJR	2	4	8	12	18	26	32	44	54	64	74	90
LB-PW	2	–	–	–	–	–	40	–	–	–	–	–
UB-PW	2	4	12	16	30	36	56	64	90	100	132	144

Finally, Post and Woeginger conjecture that the upper bound on $B_{min}(TT_n)$ for any even n and any timetable TT_n can be improved to $\frac{1}{6}n(n-1)$. However, we are able to obtain a counterexample with $n = 8$ to this conjecture by using a simple 2 phase approach. Phase 1

consists of a basic CP model for generating timetables [26] and in Phase 2 we solve the break minimization problem by using the IP model presented in [49]. The procedure iterates between the two phases and, for each number of teams n, we are able to obtain the lower bounds displayed in Table 2 within 15 minutes of computation time. Table 2 also displays the upper bounds conjectured by Post and Woeginger and we see that, for $n = 8$, our lower bound exceeds the conjectured upper bound.

Table 2: Lower bound for $\max_{TT_n} B_{min}(TT_n)$ and conjectured upper bound.

n	4	6	8	10	12	14	16	18	20	22	24	26
LB-RT	2	4	12	14	20	26	32	40	–	–	–	–
UB-Conj	2	5	9	15	22	30	40	51	63	77	92	108

The minimum break problem was motivated by the scheduling approach used by Régin [41] and Trick [49] but it only solves half the problem since it requires a given timetable. The first part of this approach regarding the timetabling problem has been considered by Henz et al. [26]. They use variables o_{is} to represent the opponent of team i in slot s and they formulate the problem using the global CP constraints *alldifferent* and *one-factor*.

$$\begin{aligned} &\textit{alldifferent}(o_{i1}, \ldots, o_{in-1}) \qquad \forall i \in 1, \ldots, n, \\ &\textit{one-factor}(o_{1s}, \ldots, o_{ns}) \qquad \forall s \in 1, \ldots, n-1. \end{aligned}$$

The *alldifferent* constraint is satisfied when each of the variables $o_{i1}, \ldots, o_{in-1}$ is instantiated to a unique value and the *one-factor* constraint is satisfied when $o_{is} \neq i \ \forall i \in 1, \ldots, n$ and $o_{is} = j$ implies that $o_{js} = i \ \forall i, j \in 1, \ldots, n$.

The two constraints make sure that any feasible solution constitutes a timetable for a single round robin tournament. However, since these are CP constraints they can be implemented in more than one way and the choice of propagation technique used for each of the constraints may have a great impact on the size of the search tree and the computation time used to solve the problem. Henz et al. provide an extensive analysis of propagation techniques to obtain guidelines for choosing the most effective solution method. In the analysis it is concluded that the propagation techniques used for the *alldifferent* constraint should obtain *arc-consistency* (see Hooker [27]) since this will reduce both the search tree and the runtime when compared to weaker consistency techniques. For the *one-factor* constraint, three propagation techniques are considered.

1. Arc-consistent propagation with respect to the constraints:

$$\begin{aligned} &o_{is} \neq i \qquad i = 1, \ldots, n, \ \ s = 1, \ldots, n-1 \\ &o_{o_{is}s} = i \qquad i = 1, \ldots, n, \ \ s = 1, \ldots, n-1 \end{aligned}$$

2. Arc-consistent propagation with respect to the constraints:

$$\begin{aligned} &o_{is} \neq i \qquad i = 1, \ldots, n, \ \ s = 1, \ldots, n-1 \\ &o_{o_{is}s} = i \qquad i = 1, \ldots, n, \ \ s = 1, \ldots, n-1 \\ &\textit{alldifferent}(o_{1s}, \ldots, o_{ns}) \qquad s = 1, \ldots, n-1 \end{aligned}$$

3. Arc-consistent propagation with respect to the constraints:

$$\textit{one-factor}(o_{1s}, \ldots, o_{ns}) \qquad \forall s \in 1, \ldots, n-1.$$

The first of the three propagation techniques leads to poor performances but, by adding the redundant *alldifferent* constraint in the second technique, much better results are obtained. The computational tests show that, when a pattern set is given, the second technique obtains the best results. The additional time used to obtain arc consistency for the *one-factor* constraint in the third technique outweights the time reduction achieved by the reduction in the search tree. However, when no pattern set is given, the third propagation technique obtains the best results. Notice that the propagation techniques are more efficient when a pattern set is given since the techniques are able to take advantage of the restrictions enforced by the pattern set.

Trick [50] is able to show why the second propagation technique works better than the third when the pattern set is given. Let D_i be the set of feasible opponents for team i in a given slot s. Then D_i is said to be bipartite if the set of teams can be divided into two sets X and Y such that

$$\begin{aligned} &|X| = |Y| = \frac{n}{2} \\ &i \in X \Rightarrow D_i \subseteq Y \\ &i \in Y \Rightarrow D_i \subseteq X \end{aligned}$$

Theorem 5 (Trick [50]) *For a given slot s, if the D_i are bipartite, then arc-consistency for the constraints*

$$\begin{aligned} &o_{is} \neq i \qquad i = 1, \ldots, n \\ &o_{o_{is}s} = i \qquad i = 1, \ldots, n \\ &\text{alldifferent}(o_{1s}, \ldots, o_{ns}) \end{aligned}$$

implies arc-consistency for the constraint

$$\text{one-factor}(o_{1s}, \ldots, o_{ns})$$

D_i is always bipartite when the pattern set is given, since the teams can be divided into the set of teams playing home and the set of teams playing away. This implies that, when the pattern set is determined before we find a timetable, there is no point in using the third propagation technique since arc-consistency for the *one-factor* constraint is obtained by the second technique and it requires less computation time. In the paper Trick provides numerous comparisons of CP and IP models for solving sports scheduling problems. These include tightly constrained timetables, schedules with home-away restrictions and schedules for more than one division. The conclusion is that IP in general performs best when an objective value is considered, while CP is best at handling the feasibility problems. However, at a few feasibility problems, IP outperforms the CP model since the propagation techniques were unable to recognize infeasibility.

Although much work has concentrated on the break minimization problem, some of the recent papers on practical sports scheduling applications find pattern sets before timetables. This approach relies on good pattern sets in the first phase but finding a characterization of feasible pattern sets is still an open problem. However, Miyashiro et al. [35] present a necessary condition for feasible pattern sets and show that the condition characterizes feasible pattern sets with a minimum number of breaks for schedules with up to 26 teams. For a subset of teams $\hat{T} \subseteq T$ they let the functions $A(\hat{T}, s)$ and $H(\hat{T}, s)$ return the number of away games and home games $\hat{T}$ plays in slot s. The necessary condition can then be stated as follows.

$$\sum_{s \in S} \min\{A(\hat{T}, s), H(\hat{T}, s)\} \geq \frac{|\hat{T}|(|\hat{T}| - 1)}{2} \qquad \forall \hat{T} \subseteq T$$

The reasoning behind the condition is that any subset of teams $\hat{T}$ must play $\frac{|\hat{T}|(|\hat{T}|-1)}{2}$ games in a single round robin tournament and, in any slot s, the teams cannot play more than $\min\{A(\hat{T},s), H(\hat{T},s)\}$ mutual games. Miyashiro et al. also show that, for pattern sets with a minimum number of breaks and no more than 26 teams the condition is both necessary and sufficient. In [35] it is shown that, whether a given pattern set with a minimum number of breaks satisfies the condition can be checked in polynomial time.

Croce and Oliveri [12] schedule the Italian soccer league and again this is a problem with a lot of additional constraints. Each team is assigned to one of two concurrent TV networks and the chosen TV network holds the rights to all the home games of the particular team. This means that the schedule should be balanced with respect to TV coverage such that both TV networks have a proportional part of the home games in each slot. Furthermore, the league contains teams sharing stadium and therefore complementary constraints must be imposed. The problem is solved by a 3-phase approach but all four decomposition steps are actually used since all patterns with no more than 4 breaks are generated before solving Phase 1. Phase 1 corresponds to Step 2, Phase 2 corresponds to Step 3 and Phase 3 corresponds to Step 4. All phases are solved by IP models and to obtain a good solution, the phases are solved iteratively according to the following scheme.

1. 200 pattern sets are generated.
2. For each generated pattern set a feasible timetable is found if possible.
3. For each generated feasible timetable, teams are allocated to placeholders.

The solution method is able to generate a number of high quality schedules and the authors note that preliminary contacts with the Italian Football (soccer) League are ongoing.

Rasmussen and Trick [39] propose another iterative approach using logic-based Benders decomposition called a pattern generating Benders approach (PGBA). This is a 4-phase approach in which Phase 1 generates patterns, Phase 2 uses an IP model to find a pattern set from the generated patterns, Phase 3 checks feasibility of the pattern set and assigns teams to placeholders and, finally, Phase 4 generates a timetable using a CP model. The resemblance to Benders decomposition comes from a number of feasibility checks in Phase 3. In case one of these checks prove the pattern set to be infeasible, a logic-based Benders cut is added to the IP model from Phase 2 and the algorithm returns to Phase 2. This iterative process continues until a feasible pattern set has been found or the IP model from Phase 2 becomes infeasible. In the first case, a corresponding timetable is found in Phase 4 and the algorithm stops. In the second case, we return to Phase 1 and generate additional patterns since Phase 1 only generates a subset of the feasible patterns initially. The algorithm continues until an optimal solution has been found or infeasibility has been proved. The computational results show that the PGBA leads to significant reductions in computation times for hard instances.

Subsequently, Rasmussen [38] has used the PGBA to schedule a triple round robin tournament for the best Danish soccer league. In the original presentation of the PGBA only place constraints were considered but numerous constraints are present in the practical application. These include constraints relating to the timetable, which makes the problem harder to solve since the subproblem becomes an optimization problem instead of a feasibility problem. Therefore not only feasibility cuts but also optimality cuts must be added to the master problem. However, the modified PGBA is able to obtain very good solutions in short time and it has been used for scheduling the 2006/2007 season of the Danish soccer league.

Recently, Bartsch et al. [4] have presented a new approach based on renewable resources for solving sports scheduling problems. They consider the problems of scheduling the German and the Austrian soccer leagues. Again various constraints must be taken into account but

in contrast to previous methods this is done by using partially renewable resources. Bartsch et al. present models based on this technique for both the German and the Austrian leagues and they develop a specialized heuristic 3-phase approach for solving the problem. In this approach, Phase 1 generates both a pattern set and a timetable with placeholders, Phase 2 assigns teams to placeholders and Phase 3 determines the exact date for each team since each slot covers more than one day. The approach has been used in practice in both Germany and Austria.

Knust and von Thaden [28] use a resource based method to obtain balanced home/away assignments for a given timetable. They define a neighbourhood for a given home/away assignment and show that all balanced pattern sets are connected. This means that it is possible to move from one balanced assignment to any other using a finite number of steps. Knust and von Thaden also consider preassignments meaning that some home-away assignments have been fixed in advance. They show that an balanced pattern set respecting these preassignments can be found in polynomial time if it exists.

A general approach for using resource-based models is presented by Drexl and Knust [13]. In their paper they show how various constraints can be modelled using resources and they are working on corresponding solution methods.

5 Minimizing Travel Distance

The minimization of travel distance becomes relevant when teams travel from one away game to the next without returning home. In this setup huge savings can be obtained when long trips are applied and teams located close together are visited on the same trip.

The interest in minimizing travel distances arose from the increasing travel costs due to the oil crises in the 1970's. This led to a request for efficient solution methods capable of finding good solutions for practical applications and a number of papers on distance minimization has appeared since 1976. In 2001 Easton, Nemhauser, and Trick [14] proposed the *traveling tournament problem* and this problem has received most of the attention concerned with minimizing travel distances since then. In the following two sections we will give an outline of the papers applied for practical applications and the papers focusing on the traveling tournament problem, respectively.

5.1 Practical Applications

Campbell and Chen [10] presented the first paper considering the problem of scheduling a basketball conference of ten teams. This is a relaxed double round robin tournament and the teams are allowed to play at most two consecutive away games without returning home. To solve the problem, a 2-phase approach is applied. In Phase 1, the optimal trips for each team are derived and the authors show that, for a tournament with an even number of teams, it is equivalent to pair the teams two and two such that the distances between the paired teams are minimized. Figure 8 shows why this holds true. Each node in the graph corresponds to a team and we want to minimize the total travel distance for team i. Team i travels at least once from or to all other teams (the dotted edges) and hence these edges can be discarded. Furthermore, the number of trips of length 2 must be maximized when minimizing the travel distance. This means that the optimal solution corresponds to a pairing of the teams (the remaining edges) which minimizes the total distance between the paired teams. This pairing is independent of the team for which we minimize the travel distance.

In Phase 2, the optimal pairing is translated into a number of feasible sequences using a constructive approach. This approach takes all the constraints into account and the result is an optimal schedule which minimizes the total travel distance.

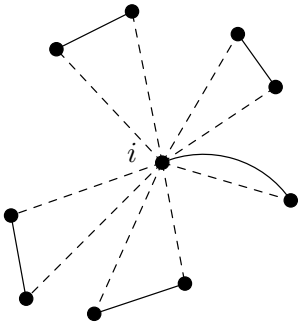


Figure 8: Graph showing the optimal trips for team i [10].

Ball and Webster [2] solve a similar scheduling problem for a basketball conference in their paper from 1977. They first model the problem using an IP formulation but the problem is too large to solve and, instead, a heuristic solution method very similar to the method by Campbell and Chen is developed.

The same year Cain, Jr. [9] presents a heuristic approach for scheduling major league baseball. The league consists of 12 teams partitioned into two divisions and each team plays 162 games making this problem one of the largest in the literature. Furthermore, a very large number of constraints and considerations are described which makes the problem even harder. The solution method is a constructive approach decomposing the season into three phases. For each phase, a pattern set is generated and, given the pattern set, an optimal timetable is subsequently found by using a computer.

In 1980, Bean and Birge [5] return to a basketball instance since they schedule the tournament for the national basketball association (NBA). As Ball and Webster, they first formulate the problem using IP but again the problem becomes too large to solve in reasonable time. Instead, they use a heuristic 2-phase approach resembling the approaches used by Campbell and Chen and Ball and Webster. However, in this problem the teams are allowed to play five consecutive away games and this relaxation makes the problem substantially harder. Furthermore, a large number of place constraints are present since the venues are used for other purposes. In Phase 1, a heuristic method is used to minimize the travel distance for each team individually. Due to the longer trips, it is no longer possible to use the "pairing approach" from the earlier methods. In Phase 2, the trips are scheduled one by one starting with the longest travel distance. The trips are scheduled in order to cover most of the home game requests and, in case a trip cannot be scheduled, it is divided into partial trips. After a feasible solution has been obtained, a switching algorithm is applied to improve the solution.

In 1991, Ferland and Fleurent [17] present a support system to help scheduling the National Hockey League (NHL). This is a relaxed tournament with 21 teams, it is divided into 2 conferences and each conference is divided into 2 divisions. The problem contains a number of constraints such as place constraints, restrictions on how often teams can play, restrictions on the minimum time between two games with the same opponents and restrictions on the traveling distances. The problem is modelled mathematically but the size of the problem makes it impossible to solve. Instead, a number of procedures are presented which can be used while the schedule is created manually. After this paper, the NHL decided to expand the league from 21 teams to 24 teams and Fleurent and Ferland [18] presented an IP model for deciding the number of games played between the four divisions.

Russell and Leung [44] considered a baseball league in 1994 with eight teams divided into

two divisions. The problem is a compact scheduling problem consisting of three segments: first a double round robin tournament for each division, then a double round robin tournament for the entire league and, finally, another double round robin tournament for each division. They apply a 2-Phase approach generating schedules for placeholders in Phase 1 and assigning teams to placeholders in Phase 2. Due to the structure of the tournament, it is possible to solve Phase 2 using total enumeration within reasonable time and Phase 1 is solved using an exchange heuristic. A feasible schedule is obtained and from this schedule new schedules are obtained by exchanging the slots. Furthermore, the number of consecutive away slots is limited to two, which means that the pairing technique from [2, 10] can be applied. They use this method to obtain a new kind of schedule with more variation compared to the traditional schedule format. However, the new schedule is rejected since it allows byes and it increases travel distance.

In the paper, they note that minimizing travel distance is correlated to maximizing the number of breaks and they prove the following theorem.

Theorem 6 (Russell and Leung [44]) *For a round robin tournament with an even number of teams $n \geq 6$ where each team can play no more than two consecutive home or two consecutive away games, the maximum number of breaks is strictly less than $n(\frac{n}{2} - 1)$.*

The first metaheuristic solution method is applied by Costa [11] in 1995. It is an evolutionary tabu search algorithm combining the mechanisms of genetic algorithms and tabu search and it is used to schedule the NHL also considered by Ferland and Fleurent [17]. The algorithm consists of three phases which are used repeatedly after initial schedules have been obtained. The initial population of schedules is generated by an algorithm similar to the one used in [17] and the road trips are built sequentially. The reproduction phase assigns a probability for each schedule to be reproduced. The probability is monotonically decreasing with respect to the number of violated constraints. The crossover phase contains the evolutionary part of the algorithm since it generates new schedules from existing schedules and the tabu search face contains a traditional tabu. The neighbourhood of the tabu search consists of all the schedules that can be obtained by moving a single game from one day to another.

Recently, two papers have appeared on minimizing travel distance for a practical application. The first is by Voorhis [54] and once more college basketball is considered. The application is a double round robin tournament with 10 teams allowing trips of length 2 (called *travel swings*). The problem is formulated as an IP model assigning games to slots and it is solved using a depth first branching algorithm. The algorithm starts with assigning trips of length two to slots and afterwards the remaining games are scheduled. For comparison the IP model is also solved using CPlex but no feasible solutions were obtained within 15 hours of CPU time. In contrast, the developed algorithm found 9 schedules in 1.33 hours of CPU time.

The second paper, by Wright [62], considers the national basketball league of New Zealand. This is a relaxed double round robin tournament with 10 teams and trips of length two are allowed. To solve the problem, a subcost-guided simulated annealing algorithm and the objective function reflects the number of violated requests. The paper gives a thorough comparison of variations of the algorithm and concludes that it is advantageous to keep a certain structure at the beginning of the search but relaxing the structural constraints during the search.

5.2 The Traveling Tournament Problem

Easton et al. [14] presented the *traveling tournament problem* (TTP) in 2001. The problem is motivated by the problem of scheduling major league baseball and it is formulated to capture the fundamental difficulties of minimizing the travel distance for a sports league. By using the TTP as benchmark problems, it is possible to develop and compare solution methods which,

afterwards, can be specialized for the various constraints present in practical applications. The TTP can be formulated as follows.

Definition 3 (Easton et al. [14]) *The traveling tournament problem is as follows:*

Input: *n, the number of teams; D an n by n integer distance matrix; L, U integer parameters.*

Output: *A double round robin tournament on the n teams such that*
- The number of consecutive home games and consecutive away games are between L and U inclusive, and
- The total distance travelled by the teams is minimized.

Furthermore, two additional requirements are mentioned. The first is a mirroring constraint requiring that the schedule is mirrored and the second is a no-repeater constraint requiring that two teams cannot play two games against each other in two consecutive slots. Notice that at most one of the two requirements is relevant since the no-repeater constraint is always satisfied in a mirrored schedule.

In the paper two instance classes are presented:

Circle instances (circular distance):
An instance of the circular distance TTP with n teams is obtained by generating an n-node circle graph with unit distances (distance of 1 between all adjacent nodes). The distance between two teams i and j with $i > j$ is then equal to the length of the shortest path between i and j and it equals the minimum of $i - j$ and $j - i + n$.

National league instances (NL):
The MLB consists of two leagues called the National League and the American League. In order to create small instances reflecting the actual structure of the MLB the teams of the National League were used to obtain benchmark problems with 4 to 16 teams called NL4, NL6,. . . ,NL16.

Later Urrutia and Ribeiro [53] have presented a third instance class:

Constant distance:
The constant distance instances are characterized by a distance of 1 between all teams and Urrutia and Ribeiro [53] show that, for this instance class, minimizing travel distance is equivalent to maximizing the number of breaks.

All the instance classes are presented at [51] together with the current best upper and lower bounds. The benchmark problems considered here all have $L = 1$ and $U = 3$.

Various solution methods have been presented for solving the TTP. Easton et al. [14] present a method based on the *independent lower bound* (IB), which they define to be the sum of the minimum travel distances for each team when they are considered independently. The solution method generates pattern sets with as many trips as possible and a corresponding timetable minimizing the travel distance is found afterwards. In this setup, a strengthening of the IB can be used to check optimality and, as long as this bound is below the best solution, the algorithm continues. This method is able to solve the NL4 and NL6 to optimality.

Benoist, Laburthe, and Rottembourg [6] apply a hybrid algorithm combining Lagrange relaxation and CP. The algorithm has a hierarchical architecture consisting of three components. The main component is a CP model capturing the entire problem and capable of solving the problem by itself. However, a global constraint is introduced in order to improve the bounds during the search. This global constraint corresponds to the second component and it contains

a Lagrange controller using either sub-gradient or modified gradient techniques to adjust the lagrange multipliers for the third component consisting of a perturbated subproblem for each team. The subproblem for a given team i schedules all the games associated with team i such that team i's travel distance is minimized.

Subsequently, Easton, Nemhauser, and Trick [15] present another hybrid IP/CP solution method. This is a branch and price (column generation) algorithm in which the columns correspond to tours for the teams. The master problem is a linear programming problem assigning teams to tours, while the pricing problem for generating tours is a CP problem. A parallel version of the algorithm is implemented and it is to date the only solution method which has been able to prove optimality of an instance of NL8. However, the no-repeater constraint was not imposed which means that the solution value can only be used as a lower bound for the instance found at [51].

The next approach for the TTP was a simulated annealing algorithm by Anagnostopoulos, Michel, Van Hentenryck, and Vergados [1] called TTSA. From an initial schedule found by a simple backtrack search TTSA searches for improving solutions using five kinds of moves: *SwapHomes, SwapRounds, SwapTeams, PartialSwapRounds* and *PartialSwapTeams*. By applying these moves, the structure of the schedule is destroyed but for each move a corresponding ejection chain is able to restore the structure. In this way the algorithm is able to satisfy all hard constraints during the search, whereas the soft constraints may be violated. The hard constraints include the round robin constraints while the no-repeater is considered a soft constraint. The number of violated soft constraints is incorporated in the objective function to force the algorithm towards feasible solutions. TTSA randomly selects a move and it is performed with probability 1 if it leads to an improving solution and otherwise the probability depends on the resulting increase in travel distance plus the current "temperature". The TTSA was able to improve all the current best known upper bounds for the NL instances with more than 10 teams and, in a recent paper by Hentenryck and Vergados [22], the TTSA is further refined to handle mirrored tournaments. In this paper they also use a randomized version of a hill climbing algorithm to obtain better initial schedules.

The first paper focussing solely on mirrored TTP instances is by Ribeiro and Urrutia [42] and they present a heuristic 3-phase approach for generating mirrored schedules quickly. In Phase 1 they first use the canonical schedule to obtain a timetable with placeholders and afterwards they construct a matrix of consecutive opponents. Each entry (i, j) of the matrix gives the number of times another team meets i and j consecutively and this is used in Phase 2 when teams are assigned to placeholders. A simple heuristic assigns teams located close together to placeholders who are met consecutively by many teams. Finally, Phase 3 uses two steps to obtain a pattern set. In Step 1 a constructive method generates an initial pattern set and afterwards Step 2 performs local search to improve the pattern set. Ribeiro and Urrutia also present a heuristic method combining GRASP and iterated local search (ILS) which they call *GRILS-mTTP*. The GRILS-mTTP performs a number of iterations all starting with the algorithm explained above for generating an initial schedule. Afterwards, a local search is applied to obtain a locally optimal solution and then GRILS-mTTP iterates between a perturbation procedure and a local search until some re-initialization criterion is satisfied.

Henz [24] proposes to combine large neighborhood search and CP to overcome the problem of getting away from local optima. He uses five types of moves which all relax a substantial part of the given schedule. For instance the move called *Relax rounds* does not only exchange two slots but it relaxes all variables associated with a number of slots. CP is then applied to obtain a new schedule given the partial schedule which has not been relaxed. In the paper it is noted that only preliminary results have been obtained and they are not competitive to the conventional local search techniques applied earlier.

As mentioned above Urrutia and Ribeiro [53] present the instance class with constant distances and show that minimizing travel distance for these instances is equivalent to maximizing the number of breaks. In the paper they also derive upper bounds on the number of breaks for unconstrained single round robin tournaments, equilibrated single round robin tournaments, unconstrained double round robin tournaments and double round robin tournaments with a maximum of three consecutive home games and three consecutive away games. The limit on consecutive home games and away games in the last kind resembles the bounds from the benchmark TTP instances. By separating these instances into three classes ($(n-1) \bmod 3 = 0$, $(n-1) \bmod 3 = 1$ and $(n-1) \bmod 3 = 2$) the following bounds were obtained.

$$UB_{\text{MTTP}} = \begin{cases} 14, & \text{if } n = 4, \\ 4(n^2-n)/3 - 4n + 20, & \text{if } ((n-1) \bmod 3 = 0 \text{ and } n \neq 4, \\ 4(n^2-2n)/3, & \text{if } ((n-1) \bmod 3 = 1, \\ 4(n^2/3 - n), & \text{if } ((n-1) \bmod 3 = 2. \end{cases}$$

The corresponding mirrored constant distance TTP is solved by the GRILS-mTTP presented in [42] and the algorithm is able to solve the instances with 4, 6, 8, 10, 12 and 16 teams to optimality by obtaining solutions which reach the upper bound stated above.

The constant distance TTP was also considered by Rasmussen and Trick [39] who used the PGBA discussed in Section 4.2 to solve the problem. They were able to prove optimality for all the mirrored instances with 18 teams or less and all the non-mirrored instances with 16 teams or less by using the algorithm for maximizing breaks instead of minimizing breaks. Hentenryck and Vergados [22] have also used their TTSA approach and improved the best solution for mirrored instance with 20 teams and the best solutions for the non-mirrored instances with 18-24 teams.

Lim et al. [29] apply a hybrid metaheuristic algorithm combining simulated annealing and hill-climbing for the TTP. After having found an initial schedule using beam search the algorithm iterates between two components for improving the current schedule. The first component searches for improving schedules by using simulated annealing. The moves in this component, called *conditional local jumps*, exchange sets of matches in such a way that all the constraints are satisfied. The second component applies hill-climbing for finding a better team assignment. This is done by means of local exchanges and the algorithm moves in the direction of decreasing travel distance. The fundamental idea of the overall approach is to improve the schedule when a good team assignment has been obtained and to search for a better team assignment when the schedule seems promising. The algorithm continues until no improvements have been obtained for a fixed number of iterations or until a time limit is reached. The computational results show that the algorithm is able to improve the best solutions for all the non-mirrored circular TTP instances with 10 teams or more.

As a generalization of the break minimization problem when distances are considered instead of breaks, Rasmussen and Trick [40] define the *timetable constrained distance minimization problem* (TCDMP). The problem is defined as follows:

Definition 4 (Rasmussen and Trick [40]) *Given a timetable for a double round robin tournament with n teams, a distance matrix specifying the distances between the venues and an upper bound UB on the number of consecutive home and consecutive away games, find a feasible pattern set which minimizes the total distance traveled by all teams.*

In the paper four solution methods for the problem are presented and evaluated. The method showing the best performances is a 2-phase hybrid IP/CP approach which generates all feasible patterns in Phase 1 using CP and assigns teams to patterns in Phase 2 using IP. In an extended version of the paper Rasmussen and Trick also present a new heuristic approach called the *circular traveling salesman approach* (CTSA) to solve the TTP. The CTSA first solves the traveling

salesman problem containing all the teams in a given tournament. Afterwards an instance of the circular distance TTP is then formulated with the teams ordered according to the TSP solution. To solve the circular distance TTP the solutions obtained by Lim et al. [29] are used and this gives a solution to the original TTP instance. In spite of the simpleness the CTSA is capable of obtaining solutions comparable to the beam search used in [29] for obtaining initial solutions.

6 Conclusion

This paper gives an outline of the terminology used within sports scheduling, it presents an overview of the constraint types used in the literature and it discusses the individual papers on both break minimization and distance minimization. From this survey it becomes clear that huge developments have taken place within the area. The solution methods become better at solving practical applications and the bounds for the benchmark instances of the TTP are improved continuously. Nevertheless, the area still holds great opportunities for new research. We can for example mention the following.

The most obvious challenge within the area and a great milestone to reach would be to prove optimality of the TTP instance NL8. Developing an efficient way of improving the lower bounds for the TTP problem and thereby reducing the current gap between the lower and upper bounds would also be extremely useful.

The area has proven to be very well suited for hybrid solution methods and this research should be continued in the future. Finding new ways of integrating IP and CP or metaheuristics and IP/CP would be beneficial not only to the sports scheduling area but also for the operations research and the computer science communities in general.

In the papers concerning practical applications many authors mention that it is very hard for the leagues to explicitly state all their requests. This makes it very hard to parameterize the model correctly and often the solution method has to be applied numerous times before a satisfactory solution has been obtained. Finding a way of changing parameters during the search could be a help in this process since then the solution method could continue instead of starting all over again.

References

[1] A. Anagnostopoulos, L. Michel, P. Van Hentenryck, and Y. Vergados. A simulated annealing approach to the traveling tournament problem. *Journal of Scheduling*, 9:177–193, 2006.

[2] B.C. Ball and D.B. Webster. Optimal scheduling for even-numbered team athletic conferences. *AIIE Transactions*, 9:161–169, 1977.

[3] F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36:493–513, 1988.

[4] T. Bartsch, A. Drexl, and S. Kröger. Scheduling the professional soccer leagues of Austria and Germany. *Computers and Operations Research*, 33(7):1907–1937, 2006.

[5] J.C. Bean and J.R. Birge. Reducing travelling costs and player fatigue in the national basketball association. *Interfaces*, 10(3):98–102, 1980.

[6] T. Benoist, F. Laburthe, and B. Rottembourg. Lagrange relaxation and constraint programming collaborative schemes for traveling tournament problems. In *Proceedings CPAIOR'01, Wye College (Imperial College), Ashford, Kent UK*, 2001.

[7] D.C. Blest and D.G. Fitzgerald. Scheduling sports competitions with a given distribution of times. *Discrete Applied Mathematics*, 22:9–19, 1988/89.

[8] E.K. Burke, D. de Werra, J.D. Landa Silva, and C. Raess. Applying heuristic methods to schedule sports competitions on multiple venues. In E.K. Burke and M.A. Trick, editors, *Proceedings PATAT 2004*, pages 451–456, 2004.

[9] W.O. Cain, Jr. The computer-assisted heuristic approach used to schedule the major league baseball clubs. In S.P. Ladany and R.E. Machol, editors, *Optimal Strategies in Sports*, volume 5 of *Studies in Management Science and Systems*, pages 32–41. North-Holland/American Elsevier, 1977.

[10] R.T. Campbell and D.S. Chen. A minimum distance basketball scheduling problem. In R.E. Machol, S.P. Ladany, and D.G. Morrison, editors, *Management Science in Sports*, volume 4 of *Studies in the Management Sciences*, pages 15–26. North-Holland Publishing Company, 1976.

[11] D. Costa. An evolutionary tabu search algorithm and the NHL scheduling problem. *INFOR*, 33:161–178, 1995.

[12] F.D. Croce and D. Oliveri. Scheduling the Italian football league: an ILP-based approach. *Computers and Operations Research*, 33(7):1963–1974, 2006.

[13] A. Drexl and S. Knust. Sports league scheduling: Graph - and resource - based models. *Omega (to appear)*.

[14] K. Easton, G. Nemhauser, and M. Trick. The traveling tournament problem: Description and benchmarks. In T. Walsh, editor, *Principles and Practice of Constraint Programming - CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 580–585. Springer Berlin / Heidelberg, 2001.

[15] K. Easton, G. Nemhauser, and M. Trick. Solving the traveling tournament problem: a combined integer programming and constraint programming approach. In E. Burke and P. De Causmaecker, editors, *Practice and Theory of AutomatedTimetabling IV*, volume 2740 of *Lecture Notes in Computer Science*, pages 100–109. Springer Berlin / Heidelberg, 2003.

[16] M. Elf, M. Jünger, and G. Rinaldi. Minimizing breaks by maximizing cuts. *Operations Research Letters*, 31:343–349, 2003.

[17] J.A. Ferland and C. Fleurent. Computer aided scheduling for a sport league. *INFOR*, 29: 14–25, 1991.

[18] C. Fleurent and J.A. Ferland. Allocating games for the NHL using integer programming. *Operations Research*, 41(4):649–654, 1993.

[19] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42: 1115–1145, 1995.

[20] J.-P. Hamiez and J.-K. Hao. Solving the sports league scheduling problem with tabu search. In A. Nareyek, editor, *Local Search for Planning and Scheduling*, volume 2148 of *Lecture Notes in Computer Science*, pages 24–36. Springer Berlin / Heidelberg, 2001.

[21] J.-P. Hamiez and J.-K. Hao. A linear-time algorithm to solve the sports league scheduling problem (prob026 of CSPLib). *Discrete Applied Mathematics*, 143:252–265, 2004.

[22] P.V. Hentenryck and Y. Vergados. Traveling tournament scheduling: A systematic evaluation of simulated annealling. In J.C. Beck and B.M. Smith, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 3990 of *Lecture Notes in Computer Science*, pages 228–243. Springer Berlin / Heidelberg, 2006.

[23] M. Henz. Scheduling a major college basketball conference - revisited. *Operations Research*, 49:163–168, 2001.

[24] M. Henz. Playing with constraint programming and large neighborhood search for traveling tournaments. In E. Burke and M. Trick, editors, *Proceedings PATAT 2004*, pages 23–32, 2004.

[25] M. Henz. Constraint-based round robin tournament planning. In D. De Schreye, editor, *Proceedings of the International Conference on Logic Programming, Las Cruces, New Mexico*, pages 545–557. MIT Press, 1999.

[26] M. Henz, T. Müller, and S. Thiel. Global constraints for round robin tournament scheduling. *European Journal of Operational Research*, 153:92–101, 2004.

[27] J.N. Hooker. *Logic-based methods for optimization combining optimization and constraint satisfaction*. Wiley Interscience, 2000.

[28] S. Knust and M. von Thaden. Balanced home-away assignments. *Discrete Optimization*, 3 354-365, 2006.

[29] A. Lim, B. Rodrigues, and X. Zhang. A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *European Journal of Operational Research*, 174(3): 1459–1478, 2006.

[30] K. McAloon, C. Tretkoff, and G. Wetzel. Sports league scheduling. In *Proceeding of the 3rd ILOG Optimization Suite International Users Conference, Paris*, 1997.

[31] E. Mendelsohn and A. Rosa. One-factorizations of the complete graph - a survey. *Journal of Graph Theory*, 9:43–65, 1985.

[32] R. Miyashiro and T. Matsui. Round-robin tournaments with a small number of breaks, 2003. URL `citeseer.ist.psu.edu/miyashiro03roundrobin.html`.

[33] R. Miyashiro and T. Matsui. A polynomial-time algorithm to find an equitable home-away assignment. *Operations Research Letters*, 33:235–241, 2005.

[34] R. Miyashiro and T. Matsui. Semidefinite programming based approaches to the break minimization problem. *Computers & Operations Research*, 33(7):1975–1982, 2006.

[35] R. Miyashiro, H. Iwasaki, and T. Matsui. Characterizing feasible pattern sets with a minimum number of breaks. In E. Burke and P. De Causmaecker, editors, *Practice and Theory of Automated Timetabling IV*, volume 2740 of *Lecture Notes in Computer Science*, pages 78–99. Springer Berlin / Heidelberg, 2003.

[36] G.L. Nemhauser and M.A. Trick. Scheduling a major college basketball conference. *Operations Research*, 46(1):1–8, 1998.

[37] G. Post and G.J. Woeginger. Sports tournaments, home-away assignments, and the break minimization problem. *Discrete Optimization*, 3(2):165–173, 2006.

[38] R.V. Rasmussen. Scheduling a triple round robin tournament for the best Danish soccer league. *Working Paper no. 2006/1, Department of Operations Research, University of Aarhus*, 2006. URL `http://www.imf.au.dk/publs?id=596)`.

[39] R.V. Rasmussen and M.A. Trick. A Benders approach for constrained minimum break problem. *European Journal of Operational Research*, 177(1):198–213, 2007.

[40] R.V. Rasmussen and M.A. Trick. The timetable constrained distance minimization problem. In J.C. Beck and B.M. Smith, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 3990 of *Lecture Notes in Computer Science*, pages 167–181. Springer Berlin / Heidelberg, 2006.

[41] J.-C. Régin. Minimization of the number of breaks in sports scheduling problems using constraint programming. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 57:115–130, 2001.

[42] C.C. Ribeiro and S. Urrutia. Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research*, 179(3):775–787, 2007.

[43] A. Rosa and W.D. Wallis. Premature sets of 1-factors or how not to schedule round-robin tournaments. *Discrete Applied Mathematics*, 4:291–297, 1982.

[44] R.A. Russell and J.M.Y. Leung. Devising a cost effective schedule for a baseball league. *Operations Research*, 42(4):614–625, 1994.

[45] R.A. Russell and T.L. Urban. A constraint programming approach to the multiple-venue, sport-scheduling problem. *Computers & Operations Research*, 33:1895–1906, 2006.

[46] A. Schaerf. Scheduling sport tournaments using constraint logic programming. *Constraints*, 4:43–65, 1999.

[47] J.A.M. Schreuder. Constructing timetables for sport competitions. *Mathematical Programming Study*, 13:58–67, 1980.

[48] J.A.M. Schreuder. Combinatorial aspects of construction of competition dutch professional football leagues. *Discrete Applied Mathematics*, 35:301–312, 1992.

[49] M. Trick. A schedule-then-break approach to sports timetabling. In E. Burke and W. Erben, editors, *Practice and Theory of Automated Timetabling III*, volume 2079 of *Lecture Notes in Computer Science*, pages 242–252. Springer Berlin / Heidelberg, 2001.

[50] M. Trick. Integer and constraint programming approaches for round robin tournament scheduling. In E. Burke and P. De Causmaecker, editors, *Practice and Theory of AutomatedTimetabling IV*, volume 2740 of *Lecture Notes in Computer Science*, pages 63–77. Springer Berlin / Heidelberg, 2003.

[51] M.A. Trick. Michael trick's guide to sports scheduling. URL `http://mat.tepper.cmu.edu/sports/Instances/`.

[52] T.L. Urban and R.A. Russell. Scheduling sports competitions on multiple venues. *European Journal of Operational Research*, 148:302–311, 2003.

[53] S. Urrutia and C.C. Ribeiro. Maximizing breaks and bounding solutions to the traveling tournament problem. *Discrete Applied Mathematics*, 1541932–1938, 2006.

[54] T.V. Voorhis. College basketball scheduling with travel swings. *Computers & Industrial Engineering*, 48:163–172, 2005.

[55] D. de Werra. Geography, games and graphs. *Discrete Applied Mathematics*, 2:327–337, 1980.

[56] D. de Werra. Scheduling in sports. In P. Hansen, editor, *Studies on Graphs and Discrete Programming*, pages 381–395. North-Holland, Amsterdam, 1981.

[57] D. de Werra. Minimizing irregularities in sports schedules using graph theory. *Discrete Applied Mathematics*, 4:217–226, 1982.

[58] D. de Werra. Some models of graphs for scheduling sports competitions. *Discrete Applied Mathematics*, 21:47–65, 1988.

[59] D. de Werra, L. Jacot-Descombes, and P. Masson. A constrained sports scheduling problem. *Discrete Applied Mathematics*, 26:41–49, 1990.

[60] D. de Werra, T. Ekim, and C. Raess. Construction of sports schedules with multiple venues. *Discrete Applied Mathematics*, 154:47–58, 2006.

[61] R.J. Willis and B.J. Terrill. Scheduling the Australian state cricket season using simulated annealing. *Journal of the Operational Research Society*, 45(3):276–280, 1994.

[62] M.B. Wright. Scheduling fixtures for basketball New Zealand. *Computers & Operations Research*, 33:1875–1893, 2006.

[63] M.B. Wright. Timetabling county cricket fixtures using a form of tabu search. *Journal of the Operational Research Society*, 45(7):758–770, 1994.