INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 30 (2023) 3267–3295
DOI: 10.1111/itor.13290

A tutorial on graph models for scheduling round-robin sports tournaments

Celso C. Ribeiro[a,*], Sebastián Urrutia[b] and Dominique de Werra[c]

[a] *Institute of Computing, Universidade Federal Fluminense, Niterói, RJ 24210-346, Brazil*
[b] *Molde University CollegeFaculty of Logistics, Molde 6402, Norway*
[c] *Ecole Politechnique Fédérale de Lausanne, Département de Mathématiques, Lausanne CH1015, Switzerland*
E-mail: celso@ic.uff.br [Ribeiro]; sebastian.a.urrutia@himolde.no [Urrutia]; dominique.dewerra@epfl.ch [de Werra]

Received 22 January 2023; received in revised form 10 March 2023; accepted 11 March 2023

Abstract

Many sports leagues organize their competitions as round-robin tournaments. This tournament design has a rich mathematical structure that has been studied in the literature over the years. We review some of the main properties and fundamental scheduling methods of round-robin tournaments. Special attention is given to extra-mural tournaments in which each team in the competition has its venue. Teams play either at home or away in such tournaments in each round. The surge of breaks in the perfect alternation of home and away breaks is studied in detail.

Keywords: sports scheduling; tournament scheduling; round robin; circle method; combinatorial optimization; graph theory

1. Introduction

Mathematical models based on theoretical graph concepts help to formulate and solve fundamental scheduling problems. We discuss some elementary sports scheduling problems and concentrate on graph theoretical and other combinatorial structures involved in their models and solution approaches. In practical situations where tournament schedules of sports leagues have to be constructed, the organizers will have various additional requirements to consider.

The basic combinatorial models will often need extensions to produce by themselves schedules that may be used in such situations. However, the problem representation using graphs or other combinatorial structures will still be crucial to developing adequate and sometimes intricate procedures that can handle the many constraints in the real scheduling problem.

*Corresponding author.

Published by John Wiley & Sons Ltd, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main St, Malden, MA02148, USA.

The main problem in sports scheduling is tournament scheduling, in which we are interested in determining the date and venue for each game of a tournament. Applications are found in scheduling tournaments of sports such as football, baseball, basketball, cricket, handball, and ice hockey. Different exact and approximate approaches, including integer programming, constraint programming, metaheuristics, and hybrid methods, have been applied to these problems. Of particular relevance is the scheduling of round-robin tournaments, a design common to many leagues and sports. An excellent presentation of the discrete mathematics structures related to tournament construction appears in Anderson (2006). The reader is referred to Rasmussen and Trick (2008) for a survey of round-robin scheduling, to Kendall et al. (2010) for an annotated bibliography of sports scheduling problems, and to Ribeiro (2012) for a tutorial on sports scheduling. The book by Ribeiro et al. (2023) details the theory and practice of combinatorial models for scheduling sports tournaments.

This tutorial is organized as follows. The next section presents the main elements of graph theory. Section 3 presents the definition of single round-robin tournaments and the basic graph model used for their representation. It also presents the circle method for constructing timetables. Section 4 introduces premature sets and discusses how not to construct a schedule. Section 5 defines home and away games and the concept of home-away patterns to advance from timetables to schedules. It also defines the concept of breaks in the perfect alternation of home and away games on each team's schedule. Section 6 discusses the structure of schedules with a minimum number of breaks. Section 7 extends the definition of single round-robin tournaments to multiround-robin tournaments. Section 8 introduces two decomposition approaches for constructing schedules. Section 9 addresses minimizing the number of rounds where breaks occur. Fairness is another criterion for scheduling sports tournaments, and Section 10 considers schedules where all teams should have the same number of breaks. Section 11 shows that minimizing the total number of travels is equivalent to maximizing the total number of breaks. Section 12 considers balanced tournaments, in which there are shared venues that impose special requirements. Section 13 considers a scheduling problem related to assigning referees to games. The minimization of carryover effects is discussed in Section 14. Finally, Section 15 shows how to find general edge colorings. Concluding remarks are drawn in the last section.

2. Elements of graph theory

This section introduces graph theory's main definitions and results, which are the essential ingredients for understanding the models and algorithms presented in this tutorial. For more definitions, results, and applications of graph theory and algorithms, we refer the reader to Berge (1985), Gould (1988), or Diestel (2010).

A graph $G = (V, E)$ consists of a set V of nodes and a set E of edges. Each edge $e = [u, v]$ is an unordered pair of distinct nodes. The nodes $u, v \in V$ are the endpoints of edge e. An edge $e = [u, v]$ may be given an orientation, say from u to v: in this case, it becomes an arc $a = (u, v)$.

A *simple graph* has neither loops (edges whose endpoints coincide) nor parallel edges (multiple edges between the same pair of endpoints). A *multigraph* may have more than one edge connecting the same pair of nodes.

The *degree* $d_G(v)$ of a node $v \in V$ in G is the number of edges having v as an endpoint (i.e., the number of edges adjacent to v). A graph $G = (V, E)$ is *complete* if there is an edge in E

between every pair of nodes $u, v \in V$, with $u \neq v$. We denote by K_n the complete graph with $n = |V|$ nodes. The graph G is *d-regular* if $d_G(v) = d$ for every node $v \in V$. A *cycle* is a sequence of edges $[u_1, u_2], [u_2, u_3], \ldots, [u_{p-1}, u_p], [u_p, u_1]$, where $u_1, u_2, \ldots, u_{p-1}, u_p$ are nodes of V.

A graph $G = (V, E)$ is *bipartite* if its node set V may be partitioned into two subsets V_1 and V_2 such that no edge in E has both endpoints in V_1 or in V_2. If every possible edge between V_1 and V_2 exists in the graph, then G is a *complete bipartite* graph. The complete bipartite graph with n nodes in one subset and m nodes in the other is denoted by $K_{n,m}$. A graph is bipartite if and only if it has no *odd cycle* (i.e., a cycle with an odd number of edges).

A subset $F \subseteq E$ *spans* the graph $G = (V, E)$ if every node $v \in V$ is the endpoint of at least one edge in F. In this case, F is called a *factor* of G. F is an ℓ-factor of G if every node $v \in V$ is an endpoint of exactly ℓ distinct edges in F.

A subset $F \subseteq E$ such that no two edges in F have a common node is called a *matching*. In a *perfect matching* F (also called a *1-regular matching*), each node $v \in V$ is adjacent to exactly one edge in F. We notice that a perfect matching is equivalent to a 1-factor.

A partition of the edge set E into matchings is an *edge coloring* of graph G. If all matchings in an edge coloring are perfect (i.e., 1-factors), then the edge coloring is a *1-factorization*.

The minimum number of matchings in an edge coloring of graph G is its *chromatic index* $\chi'(G)$. Clearly, $\chi'(G) \geq \Delta(G)$, where $\Delta(G)$ is the largest node degree in G.

Two 1-factorizations $\mathscr{F} = \{F_1, \ldots, F_k\}$ and $\mathscr{H} = \{H_1, \ldots, H_k\}$ of G are called *isomorphic* if there exists a bijective function φ from the node set V of G onto itself such that $\{F_1^{\varphi}, \ldots, F_k^{\varphi}\} = \{H_1, \ldots, H_k\}$. Here, F_i^{φ} denotes the set of all edges $[\varphi(x), \varphi(y)]$ where $[x, y]$ is an edge in F_i. The concept of isomorphic 1-factorizations can also be extended to *isomorphic edge colorings*.

A 1-factorization is said to be *perfect* when the edges in $F_i \cup F_j$ form a *Hamiltonian cycle* for each pair of distinct 1-factors F_i and F_j.

Remark 1. In a complete graph K_{2n}, any 1-factorization is equivalent to a minimum cardinality edge coloring.

This remark highlights that 1-factorizations are a special case of edge colorings. We prefer to use the term 1-factorization. When the coloring does not have the characteristics of a 1-factorization, we use the term edge coloring.

Remark 2. Formally, there is no order on the 1-factors of the 1-factorization $\mathscr{F} = \{F_1, \ldots, F_k\}$ (or on the colors of an edge coloring). We refer to the *ordered 1-factorization* $\mathscr{F} = (F_1, \ldots, F_k)$ as an ordered sequence whenever the order in which the 1-factors (or colors) are taken is relevant.

This second remark and the notation defined therein are relevant because, in the context of sports scheduling, the order of the factors of a 1-factorization matters. By analogy, we could also refer to an ordered edge coloring.

3. Single round-robin tournaments: basic graph model

In the sports scheduling problems described in this tutorial, we assume that $2n$ teams (or competitors) form a league that plays one or more tournaments. The case of an odd number of teams may be handled by adding a dummy team to the league.

Algorithm 1. Circle method

Input: Set of teams $1, 2, \ldots, 2n$
Output: Canonical 1-factorization $\mathscr{F} = (F_1, \ldots, F_{2n-1})$
for $i = 1$ ***to*** $2n - 1$ **do**
  $F_i \leftarrow \{[i, 2n]\}$;
  for $\ell = 1$ ***to*** $n - 1$ **do**
    $F_i \leftarrow F_i \cup \{[\text{mod}^{(2n-1)}(i + \ell), \text{mod}^{(2n-1)}(i - \ell)]\}$;
  end for
end for

A *single round-robin tournament* between $2n$ teams is one in which each team plays precisely once with every other team in the tournament.

It is quite natural to associate the $2n$ teams, numbered $1, \ldots, 2n$, with the nodes of a graph. Each game to be played by teams i and j will be represented by an edge $[i, j]$.

There is a one-to-one correspondence between ordered 1-factorizations and single round-robin timetables. To construct a timetable for the games of a single round-robin tournament, we have to find an edge coloring of the edges of a complete graph K_{2n} with $2n$ nodes. Since K_{2n} is $(2n - 1)$-regular, any timetable will require at least $2n - 1$ rounds, assuming each team plays at most one game in each round. Indeed, one can always construct a timetable using exactly $2n - 1$ rounds (i.e., $\chi'(K_{2n}) = 2n - 1$). Let $\mathscr{F} = (F_1, \ldots, F_{2n-1})$ be such an edge coloring, where F_i is the 1-factor whose edges correspond to the games played in round $i = 1, \ldots, 2n - 1$. $\mathscr{F}$ is indeed a 1-factorization of K_{2n} since each team has to play exactly one of its $2n - 1$ games during each one of the $2n - 1$ rounds of the timetable.

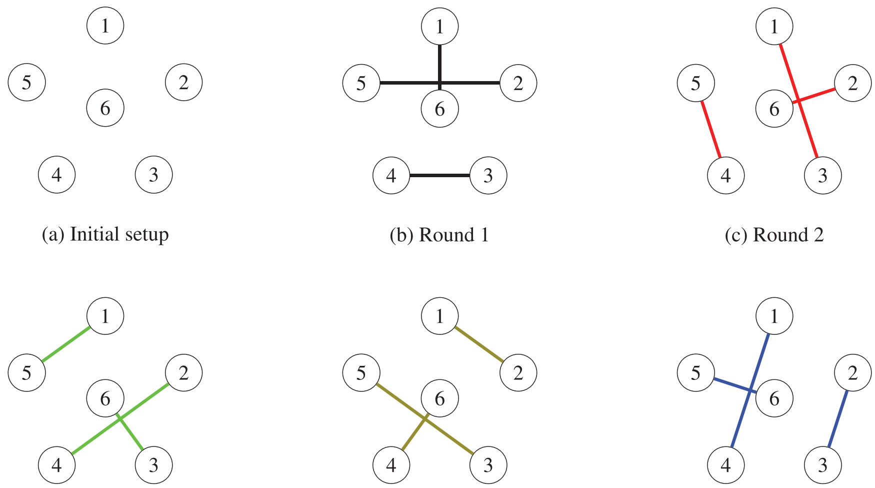
Properties and variations of edge colorings are found in Fiorini and Wilson (1977), Folkman and Fulkerson (1969), and Gabow et al. (1985). One-factorizations are discussed in Akiyama and Kano (1985) and Mendelsohn and Rodney (1994).

The *circle method* (or *polygon method*) described in Algorithm 1 was proposed originally by Kirkman (1847). It provides a simple construction for the so-called *canonical 1-factorization* $\mathscr{F} = (F_1, \ldots, F_{2n-1})$ of K_{2n}, where

$$\text{mod}^{(2n-1)}(x) = \begin{cases} x + 2n - 1, & \text{if } -(2n-2) \leq x \leq 0, \\ x, & \text{if } 1 \leq x \leq 2n - 1, \\ x - (2n - 1), & \text{if } 2n \leq x \leq 4n - 2. \end{cases} \tag{1}$$

We give in Fig. 1a graphical interpretation of the circle method. We first position the teams 1 to $2n - 1$ in the vertices of a regular polygon. For round 1, we first match the highest team of the polygon (team 1) with that in the center (team $2n$). Next, we pair each team on the left side of the polygon with that on the right side at the same height. This perfect matching defines the games of round 1. We repeat this procedure for rounds $2, 3, \ldots, 2n - 1$, as if the teams in the polygon's vertices were rotated counterclockwise at each step (or round).

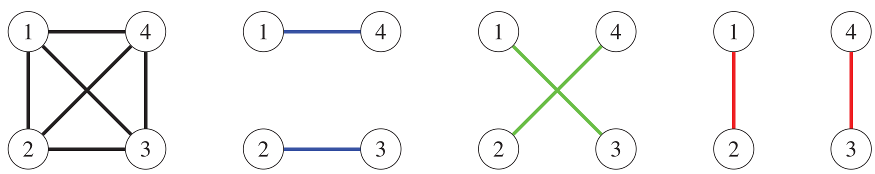
The chronological sequence of the rounds in which the games are scheduled in the 1-factorization $\mathscr{F} = (F_1, \ldots, F_{2n-1})$ induces an order on $F_1, \ldots, F_{2n-1}$. However, we shall refer to 1-factorizations without mentioning the order whenever no confusion may arise.



(a) Initial setup (b) Round 1 (c) Round 2

(d) Round 3 (e) Round 4 (f) Round 5

Fig. 1. Circle method for a tournament with $2n = 6$ teams.



(a) K_4 (b) F_1 (c) F_2 (d) F_3

Fig. 2. Canonical 1-factorization of K_4.

Table 1
Timetable corresponding to the canonical 1-factorization of Fig. 2.

Rounds		
1	2	3
[4, 1]	[4,2]	[4,3]
[2, 3]	[3,1]	[1,2]

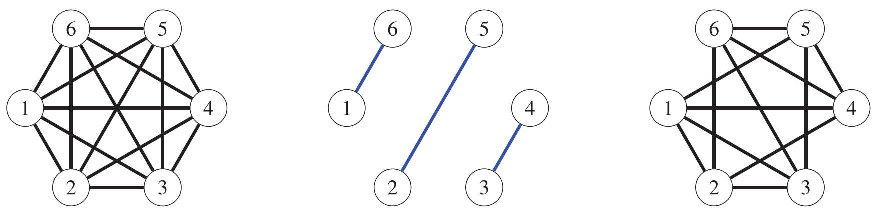
The circle method is used in some of the problems and models described later in this tutorial. Figure 2 shows a canonical 1-factorization $\mathscr{F} = (F_1, F_2, F_3)$ of K_4. In the first round, team 4 plays a game with team 1, while team 2 plays against team 3, then $F_1 = \{[4, 1], [2, 3]\}$. Similarly, $F_2 = \{[4, 2], [3, 1]\}$ and $F_3 = \{[4, 3], [1, 2]\}$.

Table 1 represents the timetable corresponding to the canonical 1-factorization of Fig. 2.

We may also represent the timetable of a single round-robin tournament of a league of $2n$ teams by a two-dimensional opponents array with $2n$ rows and $2n - 1$ columns. Each row corresponds to a team, while each column is associated with a round of the timetable. The entry in row i and

Table 2
Opponents array for the timetable of Table 1.

Teams	Rounds 1	2	3
1	4	3	2
2	3	4	1
3	2	1	4
4	1	2	3



(a) K_6 (b) F_1 (c) $K_6 - F_1$

Fig. 3. Construction on the fly of factor F_1.

column r of this array gives the opponent of team i in round r. For example, the entry in row 2 and column 3 in Table 2 denotes that team 2 meets team 1 in round 3.

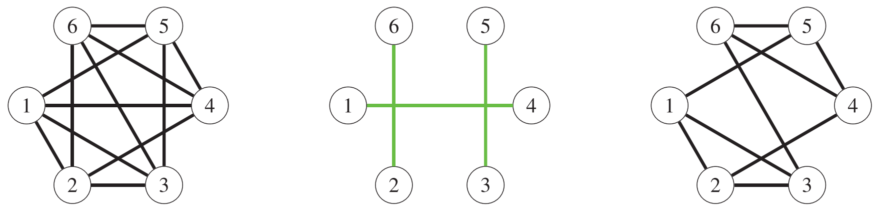
Remark 3. In leagues with an odd number of teams, there is (at least) one team that does not play a game in each round. We say that such a team has a *bye* in this round. For a single round-robin tournament represented by the complete graph K_{2n-1}, each team must play $2n-2$ games. Therefore, any timetable will require at least $2n-1$ rounds. One has $\chi'(K_{2n-1}) = 2n-1$, and one can construct a timetable with $2n-1$ rounds by starting from a timetable $\mathscr{F}$ of K_{2n} and remove all games of team $2n$ (which does not exist and may be seen as a dummy team in a league of $2n-1$ teams).

There are many more 1-factorizations for complete graphs besides the canonical ones. There is only one nonisomorphic 1-factorization for K_2, K_4, and K_6, but this number increases very fast: there are six for K_8, 396 for K_{10}, and 526,915,620 for K_{12}. We often need to use these noncanonical 1-factorizations, particularly when the problem constraints will restrict the set of rounds in which some specific games may be scheduled. We show in Section 15 another algorithm that can construct any 1-factorization of K_{2n}, including the canonical.

4. How (not) to construct a timetable: premature sets

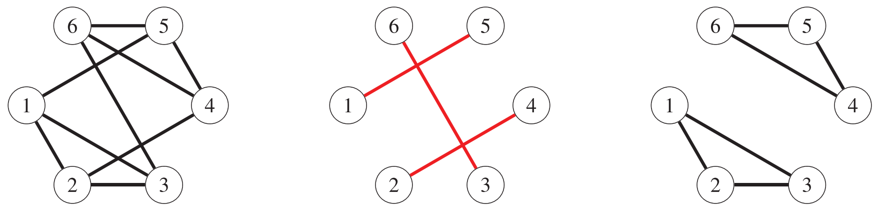
This section explains what may happen when a timetable for a single round-robin tournament of $2n$ teams has to be constructed on-the-fly, round after round.

For example, consider a tournament with $2n = 6$ teams associated with the complete graph K_6 as shown in Fig. 3a. To start, we take a set of three nonadjacent edges that form the 1-factor F_1 of K_6 in



(a) $K_6 - F_1$ (b) F_2 (c) $K_6 - F_1 - F_2$

Fig. 4. Construction on the fly of factor F_2.



(a) $K_6 - F_1 - F_2$ (b) F_3 (c) $K_6 - F_1 - F_2 - F_3$

Fig. 5. Construction on the fly of factor F_3: $\{F_1, F_2, F_3\}$ is a premature set of 1-factors.

Fig. 3b. Figure 3c illustrates the graph $K_6 - F_1$ from where the remaining factors will be obtained.

In the next step, we construct in the resulting graph $K_6 - F_1$ illustrated in Fig. 4a the 1-factor F_2 shown in Fig. 4b. The graph $K_6 - F_1 - F_2$ contains the still unused edges.

The procedure continues in the third step with the selection in the resulting graph $K_6 - F_1 - F_2$ in Fig. 5a of the 1-factor F_3 in Fig. 5b. Then, having constructed a timetable for the first three rounds defined by the three factors F_1, F_2, and F_3, we realize that the resulting graph $K_6 - F_1 - F_2 - F_3$ in Fig. 5c contains no 1-factor. Therefore, a timetable with $2n - 1 = 5$ rounds cannot be constructed, and $\{F_1, F_2, F3\}$ is a so-called premature set of 1-factors.

A set of k disjoint 1-factors of K_{2n} is called *premature* if the graph obtained by removing from K_{2n} the k 1-factors has no 1-factorization.

A premature set of k disjoint 1-factors $\{F_1, \ldots, F_k\}$ is *maximal* if $K_{2n} - (F_1 \cup \ldots \cup F_k)$ has no 1-factor. In the example of a tournament with six teams described in Figs. 3–5, the premature set $\{F_1, F_2, F3\}$ was maximal.

However, we may have premature sets that are not maximal. Although we can construct a 1-factor F_{k+1} for round $k + 1$, the choice of $\{F_1, \ldots, F_k\}$ is bad in the sense that it will not be possible to construct a 1-factorization $\{F_{k+1}, \ldots, F_{2n-1}\}$ for the remaining graph.

Several authors have studied premature sets; see, e.g., Rosa and Wallis (1982), Rees and Wallis (1991), and Caccetta and Mardiyono (1992) to understand, for instance, at which stage of the construction of a 1-factorization of K_{2n} a wrong choice may be made.

Rees and Wallis (1991) have completely characterized the critical values of k for which a maximal premature set of k 1-factors may occur. There exists a maximal premature set of k disjoint 1-factors of K_{2n} if and only if either $2\lfloor n/2 \rfloor + 1 \leq k \leq 2n - 1$ and k is odd or $(1/3)(4n + 4) \leq k \leq 2n - 4$ and k is even. Observe that necessarily $k < 2n - 2$, since after removing $2n - 2$ disjoint 1-factors, each

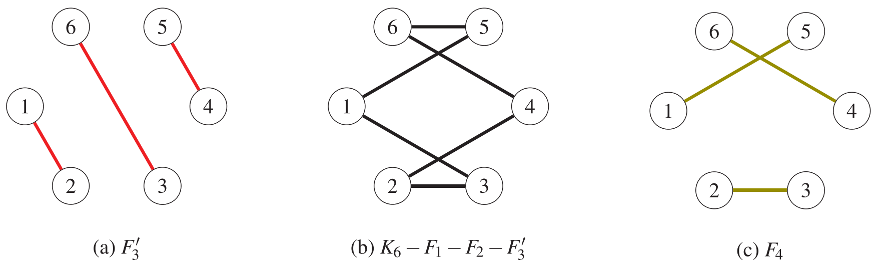


Fig. 6. Solution from a different 1-factor F_3' in the third step.

node of the remaining graph would have degree one, forming a 1-factorization of K_{2n} together with the $2n - 2$ previous 1-factors.

Knowing the values of k for which premature sets of k disjoint 1-factors may occur is not a considerable help to construct a 1-factorization of K_{2n}. However, it may indicate the procedure stage where the choice of 1-factors should be modified. In such a situation, the minimum size of a premature set (not necessarily maximal) would indicate where to restart the construction.

In the previous example of a tournament with $2n = 6$ teams, we can pick the 1-factor F_3' from $K_6 - F_1 - F_2$, as shown in Fig. 6a. The remaining graph $K_6 - F_1 - F_2 - F_3'$ appears in Fig. 6b. If the 1-factor F_4 depicted in Fig. 6c is picked from $K_6 - F_1 - F_2 - F_3'$, the remaining graph would be the 1-factor F_5, and a solution is found.

5. From timetables to schedules: home and away games

It is often the case that each team has its venue (or a stadium), and each game between teams i and j is played either in the home city of team i or the home city of team j. In the first case, we say that team i plays a *home game* and that team j plays an *away game*. It is convenient to represent this game by an arc (j, i) oriented from j to i. In the second case, if the game between teams i and j is played in the home city of team j, we have an arc (i, j) oriented from i to j.

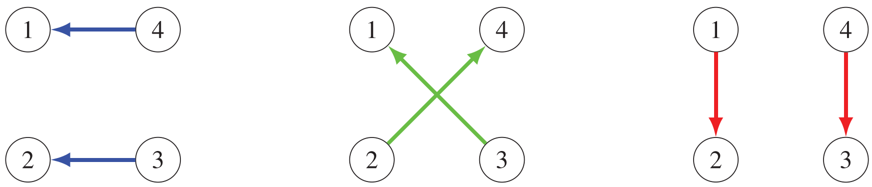
If every game in a single round-robin tournament is assigned to a venue, then each edge of K_{2n} receives an orientation and becomes an arc. Instead of an edge coloring, we now have an *oriented edge coloring*, denoted by $\overrightarrow{\mathscr{F}} = (\overrightarrow{F}_1, \ldots, \overrightarrow{F}_{2n-1})$.

For the timetable associated with K_4 in Tables 1 and 2, the edges may be oriented, e.g., as

$$\overrightarrow{F}_1 = \{(4, 1), (3, 2)\}, \overrightarrow{F}_2 = \{(2, 4), (3, 1)\}, \text{ and } \overrightarrow{F}_3 = \{(4, 3), (1, 2)\},$$

as represented in Fig. 7.

We can associate a *home-away pattern* (or a HAP) with the oriented edge coloring $\overrightarrow{\mathscr{F}} = (\overrightarrow{F}_1, \ldots, \overrightarrow{F}_{2n-1})$ and represent it by an array named *HAP* with $2n$ rows and $2n - 1$ columns, where $HAP(i, r) =$ "H" if team i plays a home game in round r, $HAP(i, r) =$ "A" if team i plays an away game in round r, or $HAP(i, r) =$ "$*$" if team i has a bye in round r, for $i = 1, \ldots, 2n$ and $r = 1, \ldots, 2n - 1$. Together, the timetable and the HAP define the tournament *schedule*.



(a) $\overrightarrow{F_1}$ (b) $\overrightarrow{F_2}$ (c) $\overrightarrow{F_3}$

Fig. 7. Oriented edge coloring for the edge coloring of Fig. 2.

Table 3
Home-away pattern associated with the oriented edge coloring of Fig. 7 (breaks are underlined).

	Rounds		
Teams	1	2	3
1	H	H	A
2	H	A	H
3	A	A	H
4	A	H	A

Using graphs to model fundamental sports scheduling problems with HAPs was first presented in de Werra (1980, 1981, 1982).

Row i of the HAP gives the *profile* of team i. Usually, one tries to alternate as much as possible the Hs and the As in the profile of each team, i.e., the home and away games played by this team. No two teams may have the same profile. To show it, we suppose that two teams had the same profiles. Since both teams would be simultaneously away or both at home, they would never be able to play against each other. Therefore, at most two teams may have perfectly alternating profiles: one that starts the tournament with a home game and another that starts with an away game.

If the profile of a team i has two consecutive entries in rounds $r-1$ and r with the same symbol (H or A), we say that team i has a *break* (of alternation) in round r. The HAP in Table 3 illustrates that teams 1 and 3 have a break in round 2. If a team plays two consecutive home games, we say it has a *home break* (as for team 1 in the table). If a team plays two consecutive away games, we say it has an *away break* (as for team 3 in the table).

We now recall some elementary properties of schedules.

Proposition 1. *Any schedule of a single round-robin tournament with* $2n$ *teams has at least* $2n-2$ *breaks.*

Proof. If an oriented edge coloring exists with $p < 2n-2$ breaks, there are at least $2n-p \geq 3$ teams without breaks in their profiles. Therefore, two teams, i and j, have the same profile. It follows that i and j can never play against each other. ■

This result shows that the oriented edge coloring of K_4 in Fig. 7 has a minimum number $2n-2 = 2$ of breaks.

Algorithm 2. Canonical procedure for edge orientation

Input: Canonical 1-factorization $\mathscr{F} = (F_1, \ldots, F_{2n-1})$
Output: Oriented canonical 1-factorization $\overrightarrow{\mathscr{F}} = (\overrightarrow{F}_1, \ldots, \overrightarrow{F}_{2n-1})$

for $i = 1$ ***to*** $2n - 1$ **do**
  if *i is even* **then**
    Edge $[i, 2n]$ becomes arc $(2n, i)$;
  else
    Edge $[i, 2n]$ becomes arc $(i, 2n)$;
  end if
  for $\ell = 1$ ***to*** $n - 1$ **do**
    if ℓ *is odd* **then**
      Edge $[\mathrm{mod}^{(2n-1)}(i+\ell), \mathrm{mod}^{(2n-1)}(i-\ell)]$ becomes arc $(\mathrm{mod}^{(2n-1)}(i+\ell), \mathrm{mod}^{(2n-1)}(i-\ell))$;
    else
      Edge $[\mathrm{mod}^{(2n-1)}(i+\ell), \mathrm{mod}^{(2n-1)}(i-\ell)]$ becomes arc $(\mathrm{mod}^{(2n-1)}(i-\ell), \mathrm{mod}^{(2n-1)}(i+\ell))$;
    end if
  end for
end for

Remark 4. One can also consider oriented edge colorings of graphs that are not complete and similarly define breaks by simply ignoring byes. There is a break (home or away) for team i in round r if entries $HAP(i, r')$ and $HAP(i, r)$, with $r' < r$, of the HAP are equal and all entries $HAP(i, r' + 1), \ldots, HAP(i, r - 1)$ are byes.

Proposition 2. *(i) For a single round-robin tournament with* $2n$ *teams, there exists a schedule with exactly* $2n - 2$ *breaks. (ii) For* $2n - 1$ *teams, there exists a schedule without breaks.*

Proof. In what follows, we give the construction of a canonical oriented edge coloring of K_{2n}. It starts from a canonical 1-factorization and gives an orientation to its edges so that teams 1 and $2n$ have no breaks and the remaining $2n - 2$ teams have one break each.

The method is called the *canonical procedure for edge orientation* and its template appears in Algorithm 2. Given a canonical 1-factorization $\mathscr{F} = (F_1, \ldots, F_{2n-1})$ obtained as described in Section 3, the procedure gives an orientation $\overrightarrow{F}_i$ to each 1-factor F_i, for $i = 1, \ldots, 2n - 1$, creating an oriented canonical 1-factorization $\overrightarrow{\mathscr{F}} = (\overrightarrow{F}_1, \ldots, \overrightarrow{F}_{2n-1})$. Table 4 illustrates an example for $2n = 6$.

We observe that team i has a unique break for any $2 \leq i \leq 2n - 1$. It occurs either in the round of game $(i, 2n)$ (if i is odd) or in the round following game $(2n, i)$ (if i is even). Teams 1 and $2n$ have no breaks. Therefore, for obtaining an oriented edge coloring of K_{2n-1}, we omit all games involving team $2n$, which gives a schedule in $2n - 1$ rounds without breaks. ■

Table 4
Canonical oriented 1-factorization of K_6. The arcs in red are oriented from node i to $2n$ (line 5) and from node $i-\ell$ to $i+\ell$ (line 11), while the arcs in blue are oriented from node $2n$ to i (line 3) and from node $i+\ell$ to $i-\ell$ (line 9) in Algorithm 2.

Canonical 1-factorization	Canonical oriented 1-factorization
$F_1 = \{[6,1],[2,5],[3,4]\}$	$\vec{F}_1 = \{(1,6),(2,5),(4,3)\}$
$F_2 = \{[6,2],[3,1],[4,5]\}$	$\vec{F}_2 = \{(6,2),(3,1),(5,4)\}$
$F_3 = \{[6,3],[4,2],[5,1]\}$	$\vec{F}_3 = \{(3,6),(4,2),(1,5)\}$
$F_4 = \{[6,4],[5,3],[1,2]\}$	$\vec{F}_4 = \{(6,4),(5,3),(2,1)\}$
$F_5 = \{[6,5],[1,4],[2,3]\}$	$\vec{F}_5 = \{(5,6),(1,4),(3,2)\}$

6. On the structure of HAPs with a minimum number of breaks

We showed in Propositions 1 and 2 that the minimum number of breaks for a single round-robin schedule with $2n$ teams is $2n-2$. We now characterize the HAPs associated with schedules with $2n-2$ breaks.

Proposition 3. *The number of home breaks equals the number of away breaks in any round of a round-robin schedule.*

Proof. There are n teams playing an away game and n teams playing a home game in each round. If, in any round, there are more home breaks than away breaks, then more teams would play home than away (and vice versa). ■

Proposition 4. *No team has more than one break in a round-robin schedule with $2n-2$ breaks.*

Proof. If a team had more than one break, there would be at least three teams without breaks, which is impossible in a round-robin tournament. ■

Proposition 5. *There are either 0 or 2 breaks in each round of any round-robin schedule with $2n-2$ breaks.*

Proof. Suppose there are $p \geq 4$ breaks in some round r and, without loss of generality, suppose they occur for teams $1, \ldots, p$. From Proposition 6, each of these p teams has a unique break, which occurs in round r. However, there are only two profiles with exactly one break in round r, one starting with a home game and the other with an away game. Since all teams must have different profiles, this situation is impossible. ■

We say that two team profiles are *complementary* if one is obtained from the other by interchanging their home and away games. We state the following corollary from the above propositions:

Corollary 1. *The HAP associated with a single round-robin tournament schedule of $2n$ teams with a minimum number of breaks consists of n pairs of complementary profiles. One pair has no break, and the $n-1$ other pairs have one break each. Each pair of complementary profiles has its breaks in a different round.*

	Rounds				
Profiles	1	2	3	4	5
1	A	H	A	H	A
2	A	A	H	A	H
3	A	H	A	H	H

(a) Three pairs of complementary profiles for a league with $2n = 6$ teams.

	Rounds				
Profiles	1	2	3	4	5
1	A	H	A	H	A
2	A	<u>A</u>	H	A	H
3	A	H	A	H	<u>H</u>
4	H	A	H	A	H
5	H	<u>H</u>	A	H	A
6	H	A	H	A	<u>A</u>

(b) Complete HAP with all $2n - 2 = 4$ breaks indicated by underlined entries.

Fig. 8. Construction of an optimal HAP for $2n = 6$ teams with $2n - 2 = 4$ breaks (breaks are underlined).

The HAP illustrated in Table 3 corresponds to a single round-robin tournament schedule of a league of four teams, where teams 2 and 4 have complementary profiles without any break. Teams 1 and 3 have complementary profiles with one break each in round 2. For the HAP in Fig. 9 with six teams, the pairs of teams with complementary profiles are 1-5, 2-6, and 3-4. The first pair of teams has no breaks, while the other pairs have one break each.

For a single round-robin tournament schedule of a league with $2n$ teams, an optimal HAP is entirely determined by providing the $n - 1$ profiles of one team in each of the pairs of complementary profiles with one break each. Alternatively, we may provide the $n - 1$ rounds where the breaks should occur (in pairs).

For example, consider a league with $2n = 6$ teams for which we want to construct a HAP with $2n - 2 = 4$ breaks occurring in pairs in rounds 2 and 5. Figure 8a shows three profiles, all starting with an away game. The first profile has no break. The second and third profiles have a break in rounds 2 and 5, respectively. The complete HAP in Fig. 8b is obtained by complementing profiles 1, 2, and 3 to obtain profiles 4, 5, and 6.

Let us now assume now that the rounds $1, 2, \ldots, 2n - 1$ are cyclically ordered, i.e., round 1 follows round $2n - 1$.

Proposition 6. *Each profile has at least one break in any cyclic HAP for* $2n$ *teams.*

Proof. Consider the profile of a team without beaks. If this team plays at home (resp. away) in the first round, it also plays at home (resp. away) in the last round since the number of rounds is odd. If the rounds are ordered cyclically, this creates a break in the first round. ■

It follows that the smallest number of breaks in a cyclic HAP is $2n$.

If we now return to the linear order of rounds $1, 2, \ldots, 2n-1$, we notice that if teams i and j have a break in round r, then reordering the rounds as $r, r+1, \ldots, 2n-1, 1, \ldots, r-1$ gives a linear HAP with precisely $2n-2$ breaks, because teams i and j have no longer a break.

Procedures for finding a HAP with minimum breaks starting from a given timetable appeared in Miyashiro and Matsui (2005). Post and Woeginger (2006) studied the worst cases that can happen when minimizing the number of breaks of a given schedule. They gave upper bounds on the minimum number of breaks.

7. Multiround-robin tournaments

Most sports leagues play either a single round-robin tournament , where each pair of teams meets once, or a double round-robin tournament , where each pair of teams meets twice. In the general case, each team plays against every other team exactly m times. The number of rounds needed to schedule all $m\binom{2n}{2}$ games is equal to $m(n-1)$, where each team plays exactly one game in each round. Double-round-robin tournaments can be organized in different manners, such as

- Unconstrained double round-robin tournaments, in which the order of the games is arbitrary.
- Tournaments with the *no-repeaters* constraint, which does not allow the two games between the same pair of teams to be played in two consecutive rounds.
- Tournaments in which a minimum number of rounds should separate the two games between the same pair of teams.
- Tournaments organized in two separate phases, each of which is an single round-robin tournament, possibly with a different timetable.
- Mirrored tournaments, in which both phases are single round-robins with the same timetables, but the venues of the two games between each pair of teams are interchanged from one phase to the other.

In the case of a mirrored double round-robin tournament, if the game between teams i and j is played at the venue of team i (resp. j) in the first phase, then it is played at the venue of team j (resp. i) in the second phase.

Therefore, one can obtain a schedule $\overrightarrow{\mathscr{H}} = (\overrightarrow{H}_1, \ldots, \overrightarrow{H}_{4n-2})$ for a mirrored double round-robin by taking $\overrightarrow{H}_i = \overrightarrow{F}_i$ for $i = 1, \ldots, 2n-1$, where $\overrightarrow{\mathscr{F}} = (\overrightarrow{F}_1, \ldots, \overrightarrow{F}_{2n-1})$ is an oriented edge coloring of K_{2n} obtained by the canonical procedure in Algorithm 2, and $\overrightarrow{H}_i = \overleftarrow{F}_{i-(2n-1)}$ for $i = 2n, \ldots, 4n-2$, where $\overleftarrow{F}_i$ is obtained by reversing all arcs of $\overrightarrow{F}_i$.

This schedule has $2n-2$ breaks in the first phase and $2n-2$ breaks in the second phase. In addition, all $2n-2$ teams who had a break in the first phase will have a break in round $2n$ (if a team has an away game in $\overrightarrow{H}_1$, then it will have a home game in $\overrightarrow{H}_{2n-1}$ and a home game in $\overrightarrow{H}_{2n}$). Thus, the schedule has $6n-6$ breaks, the minimum number of breaks for a mirrored double round-robin tournament.

This construction may lead to teams with two consecutive breaks (i.e., three consecutive home games or three consecutive away games), which might be undesirable in some situations. Such a

Algorithm 3. Modified canonical procedure for edge orientation for mirrored double round-robin tournaments

Input: Canonical 1-factorization $\mathscr{F} = (F_1, \ldots, F_{2n-1})$
Output: Oriented canonical 1-factorization $\overrightarrow{\mathscr{F}} = (\overrightarrow{F}_1, \ldots, \overrightarrow{F}_{2n-1})$

```
1  for i = 1 to 2n − 1 do
2      if (i is odd and i ≤ 2n − 5) or i = 2n − 2 then
3          Edge [i, 2n] becomes arc (i, 2n);
4      else
5          Edge [i, 2n] becomes arc (2n, i);
6      end if
7      for ℓ = 1 to n − 1 do
8          if ℓ is odd then
9              Edge [mod^(2n−1)(i + ℓ), mod^(2n−1)(i − ℓ)] becomes arc
                 (mod^(2n−1)(i + ℓ), mod^(2n−1)(i − ℓ));
10         else
11             Edge [mod^(2n−1)(i + ℓ), mod^(2n−1)(i − ℓ)] becomes arc
                 (mod^(2n−1)(i − ℓ), mod^(2n−1)(i + ℓ));
12         end if
13     end for
14 end for
```

Oriented canonical 1-factorization
$\overrightarrow{F}_1 = \{(1,6),(2,5),(4,3)\}$
$\overrightarrow{F}_2 = \{(6,2),(3,1),(5,4)\}$
$\overrightarrow{F}_3 = \{(6,3),(4,2),(1,5)\}$
$\overrightarrow{F}_4 = \{(4,6),(5,3),(2,1)\}$
$\overrightarrow{F}_5 = \{(6,5),(1,4),(3,2)\}$

	Rounds				
Teams	1	2	3	4	5
1	A	H	A	H	A
2	A	H	<u>H</u>	A	H
3	H	A	H	<u>H</u>	A
4	A	H	A	<u>A</u>	H
5	H	A	H	A	H
6	H	A	<u>A</u>	H	A

Fig. 9. Modified oriented 1-factorization of K_6 and its HAP (modified arc orientations appear in red; breaks are underlined).

situation may occur in round $2n$ of a double round-robin tournament. The teams with a break in round $2n-1$ are teams $2n-2$ and $2n-1$. We may move these breaks to earlier rounds so that no team has a break in round $2n-1$ by modifying line 2 of the original canonical procedure for edge orientation, as it appears now in Algorithm 3. This construction procedure to avoid additional breaks when concatenating the two phases appeared in de Werra (1981).

The modified oriented edge coloring of K_6 appears in Fig. 9. We notice that these modifications induce the following changes in the HAP:

- Team $2n-3$ had a break in round $2n-3$, which moved to round $2n-2$.
- Team $2n-2$ had a break in round $2n-1$, which moved to round $2n-2$.

- Team $2n - 1$ had a break in round $2n - 1$ that disappeared.
- Team $2n$ had no break before but acquired a break in round $2n - 3$.

Since there are now no breaks in round $2n - 1$, it follows that for a double round-robin tournament, there is a schedule with $6n - 6$ breaks such that no team has two consecutive breaks.

8. Constructing schedules from HAPs or from timetables

In some applications, we may have a given fixed HAP, and the question is to determine whether there is a compatible timetable assigning each game of the single round-robin tournament to the rounds $r = 1, \dots, 2n - 1$.

Such a timetable may only sometimes exist: consider, for instance, the teams represented by profiles 1, 3, and 5 in the example of Fig. 8. Teams 1 and 5 have to meet in the first round since both play home or away in every other round. Similarly, the game between teams 1 and 3 has to be scheduled in round 5. The game between teams 3 and 5 could be scheduled in either round 1 or 5. However, this cannot be done since, in both rounds, either team 3 or team 5 would already be assigned to play with team 1.

According to Corollary 1, HAPs with a minimum number of breaks have precisely two breaks in $n - 1$ of the $2n - 2$ rounds where they may occur. The example above illustrates that among the $\binom{2n-2}{n-1}$ possible HAPs with the minimum number of beaks of a single round-robin tournament of a league with $2n$ teams, there are some for which no timetable can be constructed.

An integer programming formulation can be used to determine if a given HAP (with the minimum number of breaks or not) corresponds to a timetable. This formulation was devised in Briskorn (2008) for constructing timetables from a given HAP. It is a list-edge-coloring model in which one maximizes the number of edges that can be colored.

In the opposite situation, a timetable is given, and one has to find a HAP with a minimum number of breaks.

Given a timetable for a single round-robin tournament of a league with $2n$ teams, one can find in $O(n^2)$ time whether there exists a HAP with $2n$ breaks associated with this timetable. The algorithm relies on the special structure of HAPs with a minimum number of breaks. More generally, given a timetable, it is difficult to find an associated HAP with a minimum number of breaks, i.e., with $2n - 2$ breaks. Furthermore, for a given timetable, the worst case for the minimum number of breaks is $n(n - 1)$ for n even or $(n - 1)^2$ for n odd.

9. Minimizing the number of rounds with breaks

We have seen that HAPs for single round-robin tournaments with a minimum number of breaks have exactly $n - 1$ rounds with breaks. In this section, we are interested in minimizing the number of rounds where breaks occur, which may increase the total number of breaks.

We recall that a graph is bipartite if its node set may be partitioned into two subsets V_1 and V_2 such that none of its edges has both endpoints in V_1 or in V_2. A bipartite graph has no odd cycle.

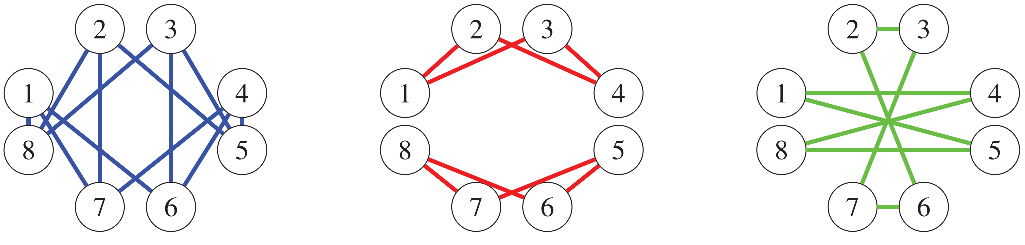


Fig. 10. Decomposition of K_8 into a minimum number of bipartite graphs.

Proposition 7. *Let $\mathscr{F} = (F_1, \ldots, F_{2n-1})$ be a 1-factorization of K_{2n} that defines a timetable for a single round-robin tournament of a league with $2n$ teams. Then, a schedule without breaks in rounds $r+1, \ldots, r+t$ can be constructed by assigning orientations to the edges of $\mathscr{F}$ if and only if $F_r \cup \ldots \cup F_{r+t}$ is a bipartite $(t+1)$-factor of K_{2n}.*

Proof. Let $\mathscr{F}$ be such a 1-factorization and assume that, for some values of r and t, $F_r \cup \ldots \cup F_{r+t}$ is a bipartite $(t+1)$-factor. Let V_1 and V_2 define a partition of the node set such that all edges in $F_r \cup \ldots \cup F_{r+t}$ have an endpoint in V_1 and the other endpoint in V_2. For every $i = r, \ldots, r+t$, we give the orientation from the node in V_1 to that in V_2 to all edges in F_i, if i is even. Otherwise, we give the orientation from the node in V_2 to that in V_1 to all edges in F_i. By giving an arbitrary orientation to the remaining edges in $\mathscr{F}$, we obtain a schedule where no breaks occur in rounds $r+1, \ldots, r+t$.

Conversely, we assume that $F_r \cup \ldots \cup F_{r+t}$ is not bipartite. Therefore, it contains an odd cycle C. In order to have no breaks between rounds $r+1$ and $r+t$, all teams must have a perfectly alternating profile from round r to round $r+t$. There are only two such profiles, one starting with a home game and the other with an away game. Two adjacent nodes in C (i.e., two teams that have to meet between rounds r and $r+t$) will have the same profile and hence will not be able to meet, which is a contradiction. Therefore, $F_r \cup \ldots \cup F_{r+t}$ is bipartite. ■

Hence, if the edge set of K_{2n} is partitioned into p subsets $M_1, \ldots, M_p$, where each M_i is a bipartite t_i-factor (with $t_1 + \cdots + t_p = 2n-1$), one may construct a schedule by giving adequate orientations to the edges. This construction is done in such a way that breaks occur only in rounds $t_1+1, t_1+t_2+1, \ldots, t_1+t_2+\cdots+t_{p-1}+1$. If we minimize p, we also minimize the number of rounds with breaks. For a league with $2n$ teams, the smallest value of p is $\lceil \log_2 2n \rceil$. In fact, one can show that the partition can be constructed in such a way that the values of t_i are either $\lceil (2n-1)/p \rceil$ or $\lfloor (2n-1)/p \rfloor$.

Proposition 8. *There is a decomposition of the edge set of K_{2n} into $p = \lceil \log_2 2n \rceil$ bipartite t_i-factors such that $-1 \le t_i - t_j \le +1$ for any $i, j \le p$.*

Figure 10 shows a decomposition for K_8 into $\lceil \log_2 8 \rceil = 3$ bipartite factors: the first of them is a 3-factor, and the other two are 2-factors.

We can also show that such a schedule with at most $2\lfloor \frac{n}{2} \rfloor \lceil \log_2 n \rceil$ breaks can be found. Related results and properties, including the proof of Proposition 8, are found in Kotzig (1972) and de Werra (1982).

Table 5
HAP for $2n = 8$ teams where each team has one away and one home break (breaks are underlined).

Teams	Rounds						
	1	2	3	4	5	6	7
1	A	$\underline{\text{A}}$	H	A	H	$\underline{\text{H}}$	A
2	A	H	$\underline{\text{H}}$	A	$\underline{\text{A}}$	H	A
3	A	H	A	$\underline{\text{A}}$	H	$\underline{\text{H}}$	A
4	A	H	A	H	$\underline{\text{H}}$	A	$\underline{\text{A}}$
5	H	$\underline{\text{H}}$	A	H	A	$\underline{\text{A}}$	H
6	H	A	$\underline{\text{A}}$	H	$\underline{\text{H}}$	A	H
7	H	A	H	$\underline{\text{H}}$	A	$\underline{\text{A}}$	H
8	H	A	H	A	$\underline{\text{A}}$	H	$\underline{\text{H}}$

10. Fairness of schedules

Fairness is another criterion for tournament scheduling. The first idea is to require that all teams have the same number of breaks. Single round-robin schedules in which every one of the $2n$ teams has one break are not more difficult to obtain than schedules with $2n - 2$ breaks. Indeed, from any schedule in which each team has one break, one can derive a schedule with $2n - 2$ breaks by a simple cyclic permutation of the rounds (if two teams have a break in round r, then taking the rounds in the order $r, r + 1, \ldots, 2n - 1, 1, 2, \ldots, r - 1$ will give a schedule with $2n - 2$ breaks).

Similarly, from any schedule with $2n - 2$ breaks, one can get a schedule with $2n$ breaks by adequately permuting the rounds cyclically.

In these schedules, one-half of the teams have an away break, while the other half have a home break. This situation sometimes is considered unfair, and for a schedule to be fair, each team should have the same number of away and home breaks.

When it is required that for each team, the number of home and away breaks must be the same, the best schedule we can have is one in which each team has exactly one home break and one away break. A HAP of this type is represented in Table 5 for $2n = 8$ teams.

There is still some unfairness in such a schedule, as we will see. Assume that some team has a larger interval between their two breaks than others. If the team considered has an away break first, this team will likely play many games with low vitality caused by the preceding away break (for two consecutive rounds, the team has not played at home). Conversely, if a team has a home break first, this team may take advantage of its first home break for a longer period.

To prevent this kind of unfairness, we look for schedules where the two breaks of each team are adjacent so that the respective advantages or disadvantages are immediately canceled out. Table 6 shows such a HAP for a league with $2n = 8$ teams. The HAP always consists of n pairs of complementary profiles for such schedules.

Knust and von Thaden (2006) considered the situation where HAPs have to be constructed so that, for each team, the numbers of home and away games differ by at most one. They suggested repairing techniques for improving unbalanced HAPs. Tanaka and Miyashiro (2016) studied HAPs for which each team has one home break and one away break. They require that the distance

Table 6
HAP for $2n = 8$ teams where the two breaks of each team are consecutive (breaks are underlined).

	Rounds						
Teams	1	2	3	4	5	6	7
1	A	A	H	H	A	H	A
2	A	H	H	A	A	H	A
3	A	H	A	A	H	H	A
4	A	H	A	H	H	A	A
5	H	H	A	A	H	A	H
6	H	A	A	H	H	A	H
7	H	A	H	H	A	A	H
8	H	A	H	A	A	H	H

between the two breaks is the same for all teams, and properties of such fair schedules were derived. Suzuka et al. (2007) also examined fair schedules in terms of breaks.

11. Travel minimization and break maximization

Generally, the $2n$ teams of a league playing a single round-robin tournament are located in different cities, and the distances between the cities are known. All teams start the tournament at home, where they should return after their last away game. In this situation, minimizing the total distance traveled by the teams may be more critical than minimizing the number of breaks.

If all distances are assumed to be equal to one, then the problem becomes the minimization of the total number of travels the teams perform. A timetable T is given, and we want to find a HAP minimizing the total number of travels. We can show that the problem amounts to maximizing the number of breaks. Before the first round, the n teams starting with an away game will travel. At the end of the tournament, all n teams with an away game will travel back home. After each round $r = 1, \dots, 2n-2$, only the teams with home breaks in the next round do not travel.

If $B(T)$ denotes the total number of breaks in the schedule based on timetable T, then the total number of home breaks is $B(T)/2$. Therefore, the total number of trips after round $2n-2$ is $2n(2n-2) - B(T)/2$. Hence, the total number of trips is

$$n + 2n(2n-2) - B(T)/2 + n = 2n(2n-1) - B(T)/2.$$

It follows that for minimizing the number of travels, we maximize the number $B(T)$ of breaks; see Urrutia and Ribeiro (2006).

A schedule with the maximum number of breaks may be obtained from a schedule minimizing the number of breaks. To do so, we start from a HAP $\mathscr{H}$ with the minimum number of breaks and create another HAP $\mathscr{H}^*$ by exchanging the home-away status of all games in the even rounds of $\mathscr{H}$, as illustrated in Fig. 11.

	Rounds				
Teams	1	2	3	4	5
1	A	H	A	H	A
2	A	$\underline{A}$	H	A	H
3	A	H	A	H	$\underline{H}$
4	H	A	H	A	H
5	H	$\underline{H}$	A	H	A
6	H	A	H	A	$\underline{A}$

(a) HAP $\mathscr{H}$ with $2n-2=4$ breaks (minimum) indicated by underlined entries

	Rounds				
Teams	1	2	3	4	5
1	A	$\underline{A}$	$\underline{A}$	$\underline{A}$	$\underline{A}$
2	A	H	$\underline{H}$	$\underline{H}$	$\underline{H}$
3	A	$\underline{A}$	$\underline{A}$	$\underline{A}$	H
4	H	$\underline{H}$	$\underline{H}$	$\underline{H}$	$\underline{H}$
5	H	A	$\underline{A}$	$\underline{A}$	$\underline{A}$
6	H	$\underline{H}$	$\underline{H}$	$\underline{H}$	A

(b) HAP $\mathscr{H}^*$ with $(2n-1)(2n-2)=20$ breaks (maximum) indicated by underlined entries

Fig. 11. Construction of a HAP $\mathscr{H}^*$ with the maximum number of breaks from a HAP $\mathscr{H}$ with the minimum number of breaks for $2n=6$ teams (breaks are underlined).

Note that for every team i and every round $r = 2, \ldots, 2n-1$, entry (i, r) is a break in HAP $\mathscr{H}^*$ if and only if it is not a break in HAP $\mathscr{H}$. Hence, $B(\mathscr{H}) + B(\mathscr{H}^*) = 2n(2n-2)$ and $\mathscr{H}$ will be a HAP minimizing the number of breaks if and only if $\mathscr{H}^*$ is HAP maximizing the number of breaks.

12. Balanced tournaments

We have assumed that each team has its venue (or stadium). This situation is not always the case, and we also have to consider models in which the $2n$ teams have to play their games in a set of shared venues. Special requirements regarding these venues (courts, fields, stadiums, locations) may also have to be considered for some sports scheduling problems.

We first consider a single round-robin tournament of a league with $2n$ teams and a collection of n venues. We would like to have a timetable as balanced as possible, i.e., in which all teams play approximately the same number of games in each venue. Since each of the $2n$ teams has $2n-1$ games to play, one would like each team to have one or two games in each of the n venues. A schedule satisfying this requirement is called a *balanced tournament*. In some cases, instead of venues, the assigned time slots in the rounds (starting times of the games) have to be balanced, assuming that no games can be played in parallel. Balanced schedules exist for any $2n \neq 4$. A simple construction is described below for the cases where $2n = 0 \pmod 3$ or $2n = 2 \pmod 3$.

	Venues		
Rounds	1	2	3
1	[6,1]	[2,5]	[3,4]
2	[6,2]	[3,1]	[4,5]
3	[6,3]	[4,2]	[5,1]
4	[6,4]	[5,3]	[1,2]
5	[6,5]	[1,4]	[2,3]

(a) Original unbalanced assignments of games to venues.

	Venues		
Rounds	1	2	3
1	[2,5]	[6,1]	[3,4]
2	[4,5]	[3,1]	[6,2]
3	[6,3]	[4,2]	[5,1]
4	[1,2]	[5,3]	[6,4]
5	[1,4]	[6,5]	[2,3]

(b) Balanced assignments of games to venues.

Fig. 12. Construction of a balanced tournament for $2n = 6$ teams (games in red had their venues changed).

We start from a canonical ordered 1-factorization $(F_1, \ldots, F_{2n-1})$ of K_{2n}. For each round $r = 1, \ldots, 2n-1$, we denote by F_r^i the edge corresponding to the game scheduled at the venue $i = 1, \ldots, n$. In each round, games are assigned to venues as they are created by the circle method in Algorithm 1. For each round $r = 1, \ldots, 2n-1$, the game between teams $2n$ and r is first assigned to venue 1. Following, the game created in each iteration $\ell = 1, \ldots, n-1$ is assigned to venue $\ell + 1$.

At this point, all teams except team $2n$ have a balanced schedule because team $2n$ is always playing at the venue $i = 1$. For every $i = 1, \ldots, n-1$, we exchange the venues associated with edges F_i^1 and F_i^{i+1}. Similarly, for every $i = n+1, \ldots, 2n-1$, we exchange the venues associated with edges F_i^1 and F_i^{2n+1-i}. We observe that this construction balances the games of every team over the n venues, i.e., all teams play twice in all but one venue and once in the remaining venue. An illustration for a league of $2n = 6$ teams appears in Fig. 12.

Balanced tournaments for $2n$ teams have been shown to exist for all values of $2n \neq 4$. To make the problem more regular for a theoretical study, instead of using a complete graph K_{2n}, one may consider the graph K_{2n}^* obtained from K_{2n} by doubling all edges of a single 1-factor. We now have the following requirements:

(a) all pairs of teams meet at least once;
(b) there are n games in each round with $2n$ participating teams;
(c) exactly one game is played in each venue in each round;
(d) each team plays two games in each of the n venues; and
(e) each pair of teams plays no more than one game in each venue.

Table 7
Balanced schedule for $2n = 10$ teams in $s = 3$ venues. Each team plays $2n - 1 = 9$ games, three of which in each venue.

	Venues		
Rounds	1	2	3
1	[1,10]	[3,6]	[2,8]
2	[1,5]	[3,7]	[2,9]
3	[2,6]	[1,8]	[3,4]
4	[2,7]	[1,9]	[3,5]
5	[3,8]	[2,4]	[1,6]
6	[3,9]	[2,10]	[1,7]
7	[2,3]	[5,8]	[6,7]
8	[5,10]	[6,9]	[7,8]
9	[6,4]	[1,3]	[8,9]
10	[7,5]	[8,4]	[9,10]
11	[8,6]	[9,5]	[4,10]
12	[7,9]	[6,10]	[4,5]
13	[8,10]	[4,7]	[1,2]
14	[1,4]	[2,5]	[3,10]
15	[4,9]	[7,10]	[5,6]

Finding a schedule meeting requirements (a)–(e) is slightly different from the balanced tournament design problem. For instance, although there is a balanced tournament for K_6, there are no schedules satisfying requirements (a)–(e) for K_6^*, as it may be seen by inspection. There is no solution either for K_4 or for K_4^*. However, a simple inductive construction was developed for K_{2n}^* when $2n$ is a power of two greater than or equal to eight. It is based on dividing the league with $2n$ teams into two disjoint subleagues. These ideas have been explored later to derive a procedure for the general case of $2n \geq 8$ teams.

In some situations, the number s of venues may be smaller than the number of games in each round of a compact schedule, i.e., $s < n$. A balanced noncompact schedule exists when $2n - 1 = 0 \bmod s$ or $n = 0 \bmod s$. As an illustration, consider a league with $2n = 10$ teams. Table 7 shows that a noncompact balanced schedule exists when the number of venues is $s = 3$. Note that $2n - 1 = 9$ is a multiple of $s = 3$. There are $n(2n - 1) = 45$ games to be scheduled in $n(2n - 1)/s = 15$ rounds.

In a more general setting, the number of available venues may vary from round to round. More precisely, we are given the number of rounds $q \geq 2n - 1$ and the number of venues s_r available in each round $r = 1, \ldots, q$. Balancing is no longer an issue, but the question becomes to determine if a schedule exists where exactly s_r games are played in round $r = 1, \ldots, q$.

The condition $\sum_{r=1}^{q} s_r = n(2n - 1)$ holds, and we assume that $s_r \leq n, r = 1, \ldots, q$. For convenience and without loss of generality, we assume that $s_1 \geq s_2 \geq \cdots \geq s_q$. This assumption does not restrict the model since the q rounds may be arbitrarily permuted.

A schedule with s_r games in each round $r = 1, \ldots, q$ always exists and can be constructed from a canonical 1-factorization of K_{2n}. For each 1-factor F_i of the ordered canonical 1-factorization $\mathscr{F} = (F_1, \ldots, F_{2n-1})$, we pick each of its edges in the order $F_i^1, \ldots, F_i^n$ and assign it to the first

	Edges			
1-factors	1	2	3	4
F_1	[8,1]	[2,7]	[3,6]	[4,5]
F_2	[8,2]	[3,1]	[4,7]	[5,6]
F_3	[8,3]	[4,2]	[5,1]	[6,7]
F_4	[8,4]	[5,3]	[6,2]	[7,1]
F_5	[8,5]	[6,4]	[7,3]	[1,2]
F_6	[8,6]	[7,5]	[1,4]	[2,3]
F_7	[8,7]	[1,6]	[2,5]	[3,4]

(a) Ordered canonical 1-factorization of K_8

	Games			
Rounds	1	2	3	4
1	[8,1]	[2,7]	[3,6]	[4,5]
2	[8,2]	[3,1]	[4,7]	[5,6]
3	[8,3]	[4,2]	[5,1]	[6,7]
4	[8,4]	[5,3]	[6,2]	
5	[7,1]	[8,5]	[6,4]	
6	[7,3]	[1,2]	[8,6]	
7	[7,5]	[1,4]		
8	[2,3]	[8,7]		
9	[1,6]	[2,5]		
10	[3,4]			

(b) Games scheduled for each round

Fig. 13. Construction of a timetable with s_r venues in round r.

round with a place still available. Figure 13 illustrates this construction for $2n = 8$ teams on 10 rounds. The number of venues available in each round is $s_1 = s_2 = s_3 = 4$, $s_4 = s_5 = s_6 = 3$, $s_7 = s_8 = s_9 = 2$, and $s_{10} = 1$. This strategy gives a schedule where each team plays at most one game in each round. Figure 13a gives the ordered canonical 1-factorization of K_8. Figure 13b displays the games scheduled for each round.

Balanced tournaments were defined and studied in Schellenberg et al. (1977). A regularized version of the problem was studied by de Werra et al. (2006) and Ikebe and Tamura (2008). In Mendelsohn and Rodney (1994), stadium-balanced tournament design was studied for an arbitrary number of stadiums. Geinoz et al. (2008) studied some constructions of balanced schedules based on partitions into subleagues. Erzurumluoglu (2018) exploited the idea of partitioning the league for constructing several types of balanced schedules.

Referees

	[8, 1]	[4, 5]	[2, 7]		[3, 6]		
		[8, 2]	[5, 6]	[1, 3]		[4, 7]	
			[8, 3]	[6, 7]	[2, 4]		[1, 5]
Rounds	[2, 6]			[8, 4]	[7, 1]	[3, 5]	
		[3, 7]			[8, 5]	[1, 2]	[4, 6]
	[5, 7]		[4, 1]			[8, 6]	[2, 3]
	[3, 4]	[6, 1]		[2, 5]			[8, 7]

Fig. 14. Room square of size $2n - 1 = 7$.

13. Tournaments with referees

We now consider another type of scheduling problem related to the assignment of referees to games. We want to construct a timetable for a single round-robin tournament of a league with $2n$ teams for which we have $2n - 1$ referees. Exactly one referee should be assigned to each game. The goal is to find a compact timetable in $2n - 1$ rounds such that each team sees every referee exactly once.

The combinatorial structure needed to model this problem is a *Room square* (of size $2n - 1$) built on a set $T = \{1, \ldots, 2n\}$ corresponding to the $2n$ teams. A Room square is a $(2n - 1) \times (2n - 1)$ array where every entry is either empty or contains an unordered pair of distinct elements $i, j \in T$, corresponding to the game between teams i and j. In addition, it satisfies the following conditions: (i) every element of T occurs exactly once in every row and exactly once in every column, and (ii) every unordered pair of elements of T appears in exactly one entry of the Room square. The original concept of s Room square appeared in Kirkman (1850) and was later formalized by Room (1955) and studied by Mullin and Wallis (1975). Construction procedures for Room squares appeared in Anderson (1997).

An example of a Room square of size $2n - 1 = 7$ is shown in Fig. 14. There are eight teams, and the timetable is represented by the nonempty entries of the Room square. Its rows correspond to the $2n - 1$ rounds, and its columns are associated with the $2n - 1$ referees. Room squares can be found for any odd number greater than or equal to seven.

We may now consider a further extension of the problem with $2n - 1$ referees by introducing $2n - 1$ venues. We require, in addition, that each team plays its $2n - 1$ games in different venues.

This extension introduces a third dimension to the Room square previously considered. We now have a *Room cube* of size $2n - 1$, i.e., a three-dimensional array with the property that each one of its two-dimensional projections is a Room square of size $2n - 1$. The extension of Room squares to Room cubes and p-dimensional Room cubes was presented in Dinitz and Stinson (2005).

A Room cube based on the Room square of Fig. 14 is shown in Fig. 15. Each nonempty entry in row r and column u of the Room cube contains a pair $[i, j]$ and a venue identification s, meaning that teams i and j play against each other in round r with referee u at the venue s.

Referees

Rounds

1 [8 ,1]	7 [4, 5]	6 [2, 7]		4 [3, 6]		
	2 [8, 2]	1 [5, 6]	7 [1, 3]		5 [4, 7]	
		3 [8, 3]	2 [6, 7]	1 [2, 4]		6 [1, 5]
7 [2, 6]			4 [8, 4]	3 [7, 1]	2 [3, 5]	
	1 [3, 7]			5 [8, 5]	4 [1, 2]	3 [4, 6]
4 [5, 7]		2 [4, 1]			6 [8, 6]	5 [2, 3]
6 [3, 4]	5 [6, 1]		3 [2, 5]			7 [8, 7]

Fig. 15. Room cube of size $2n - 1 = 7$.

14. Carryover effects

Minimizing the value of the carryover effects is another requirement often present in the construction of schedules for sports tournaments. The carryover effect has generated extensive research due to the irregularities it may create in a tournament. It was studied first in Russell (1980), where a construction for a balanced tournament was devised for the case when the number of teams is a power of two.

In this problem, we seek to avoid having many teams such that all of them play against two teams, i and j, consecutively in the same order. In other words, if we consider the sets formed by ordered pairs (i, j) of consecutive opponents of each team, we would like to have all of them pairwise disjoint. Otherwise, if this is impossible, we want to minimize the sum of the squares of the number of times each pair of consecutive opponents appears over all sets.

We define a *Latin square* as a $q \times q$ array in which every row contains the numbers 1 to q, and so does every column. The timetable in Fig. 16 illustrates an example of a league with $2n = 4$ teams. Figure 16a gives the opponents of teams 1 to 4 in each round. For convenience, we consider a cyclic timetable, assuming that round 1 follows (or is repeated after) round 3. Figure 16b displays the pairs of consecutive opponents for each team. Team 1, for example, meets consecutively with teams 4, 3, and 2. Hence, the pairs of consecutive opponents for team 1 in the cyclic schedule are (4,3), (3,2), and (2,4). We artificially introduce the pair (i, i) in the first column of the table of ordered pairs of consecutive opponents (although it does not correspond to a real pair of opponents). This table can be seen as a $2n \times 2n$ array containing $4n^2$ pairs of teams. We would like to have (if possible) all these pairs different, as it happens to be the case in Fig. 16b. If we decompose the $2n \times 2n$ array of ordered pairs into two $2n \times 2n$ arrays

	Rounds			
Teams	1	2	3	1
1	4	3	2	4
2	3	4	1	3
3	2	1	4	2
4	1	2	3	1

(a) Opponents of teams 1, 2, 3, and 4 in each round of a cyclic timetable

Team i (i,i)	Pairs of consecutive opponents of team i		
$(1,1)$	$(4,3)$	$(3,2)$	$(2,4)$
$(2,2)$	$(3,4)$	$(4,1)$	$(1,3)$
$(3,3)$	$(2,1)$	$(1,4)$	$(4,2)$
$(4,4)$	$(1,2)$	$(2,3)$	$(3,1)$

(b) Pairs of consecutive opponents of teams 1, 2, 3, and 4

1	4	3	2
2	3	4	1
3	2	1	4
4	1	2	3

1	3	2	4
2	4	1	3
3	1	4	2
4	2	3	1

(c) Two orthogonal Latin squares obtained from the above pairs of consecutive opponents

Fig. 16. Timetable with no repeating pairs of consecutive opponents.

by taking the first element i (resp. the second element j) of each ordered pair (i, j), we obtain two Latin squares. Requiring that all ordered pairs differ amounts to saying that the two Latin squares are *orthogonal*. Since orthogonal Latin squares of size six do not exist, in any timetable for a league with $2n = 6$ teams, at least two teams will consecutively meet some pair of teams.

It has nevertheless been shown that if $2n$ is a power of two, then there is a schedule in $2n - 1$ rounds such that each pair of consecutive opponents occurs precisely once.

Heuristics for the minimum carryover effects problem appeared, e.g., in Guedes and Ribeiro (2011) and Cao et al. (2022). Notably, the canonical 1-factorization of K_{2n} maximizes the value of the carryover effects. Goossens and Spieksma (2012) examined the popularity of the canonical schedule and how it was used in 25 European football (soccer) competitions for the season 2008–2009. The authors also discussed how the schedules managed the carryover effects. Lambrechts et al. (2018) showed that schedules generated for round-robin tournaments by the circle method have maximum carryover effects. Consequently, as discussed in the following section, there is a need to examine other 1-factorizations, not isomorphic to the canonical 1-factorization.

(a) Edge coloring of K_3: $\chi'(K_3) = 3$ (b) Edge coloring of K_4: $\chi'(K_4) = 3$

Fig. 17. Finding an edge coloring of K_4 (right) from an edge coloring of K_3 (left).

15. How to find general 1-factorizations?

The chromatic index $\chi'(G)$ of a graph $G = (V, E)$ was defined in Section 2. Vizing's theorem (Vizing, 1964) states that if G is a simple graph (i.e., a graph without loops or parallel edges), then its chromatic index $\chi'(G) \leq \Delta(G) + 1$, where $\Delta(G)$ is the largest degree of any node of graph G. Different constructive proofs of Vizing's theorem provide algorithms for constructing edge colorings of G using, at most, $\Delta(G) + 1$ colors. Gabow et al. (1985) presented another edge-coloring procedure that can be used for 1-factorizations.

In the particular case of complete graphs, $\Delta(K_{2n}) = 2n - 1$ and $\Delta(K_{2n-1}) = 2n - 2$. Vizing's theorem implies that $\chi'(K_{2n}) \leq \Delta(K_{2n}) + 1 = 2n$ and $\chi'(K_{2n-1}) \leq \Delta(K_{2n-1}) + 1 = 2n - 1$. It turns out that this bound is tight for K_{2n-1}, but not for K_{2n} because, as we have already seen, K_{2n} can be colored with $2n - 1$ colors.

An edge coloring of K_{2n} with $2n - 1$ colors (i.e., a 1-factorization of K_{2n}) can be obtained from an edge coloring of K_{2n-1} with the same $2n - 1$ colors that, in turn, can be built using the constructions of Vizing's theorem. If K_{2n-1} can be edge colored with $2n - 1$ colors, it is possible to obtain an edge coloring with the same number of colors for K_{2n}. First, we observe that a different color (among the $2n - 1$ colors) is missing at the edges incident to each node of K_{2n-1}. Since each color is used in exactly $n - 1$ edges of K_{2n-1}, it is missing in one node. Therefore, we can add a new node to the graph K_{2n-1} and color the new edges connecting this node to the original nodes of K_{2n-1} with the missing color of each node. Hence, an edge coloring of K_{2n} with $2n - 1$ colors is obtained. An application of this procedure is illustrated in Fig. 17.

We now show that the edges of an arbitrary graph G can be colored with $\Delta(G) + 1$ colors. The procedure starts from a graph G partially colored with no more than $\Delta(G) + 1$ colors. At each step, a new, yet uncolored edge $e_0 = [w, v_0]$ will be colored without using more than $\Delta(G) + 1$ colors. This procedure is repeated until all edges are colored when it obtains the desired edge coloring with $\Delta(G) + 1$ colors.

We denote by $Free(v)$ the set of colors missing at node v, i.e., the set of colors not used in the edges incident to node v. Since the degrees of nodes v_0 and w are smaller than or equal to $\Delta(G)$, we have $|Free(v_0)| \geq 2$ and $|Free(w)| \geq 2$.

If there is a color c that belongs to $Free(v_0) \cap Free(w)$, then we simply color edge e_o with c.

Otherwise, $Free(w) \cap Free(v_0) = \emptyset$. Then, let $\alpha_0 \in Free(v_0)$ and $\beta \in Free(w)$ be two free colors at nodes v_0 and w, respectively. Let P be a maximal chain starting from v_0 such that its edges are alternately colored with colors β and α_0. Such a chain will be called an α_0/β-chain. Two cases may arise: P does not end in node w or P ends in node w. In the first case, edge $e_0 = [w, v_0]$ may be colored and the edge coloring is augmented. This is done by exchanging the colors of the edges along the chain P and coloring edge $e_0 = [w, v_0]$ with color β that is now free in v_0.

In the second case, we assume that the chain P ends at node w. If we exchange the colors of all edges of P, then $\beta \in Free(v_0)$ but $\beta \notin Free(w)$.

Then, let v_1 be the node adjacent to w in P and $e_1 = [w, v_1]$ be the edge that is colored with α_0. At this point, the color α_0 is removed from edge e_1 and edge e_0 is colored with that same color α_0.

The problem consists of recoloring edge e_1. The same procedure applied to edge e_0 can be used to assign a color to edge e_1. If $Free(w) \cap Free(v_1)$ is also empty, then the color to be selected from $Free(v_1)$ must be different from α_0 in order to avoid cycling. This color always exists, since $|Free(v_1)| \geq 2$.

We may continue with this procedure until $Free(w) \cap Free(v_i)$ becomes not empty or the chain P does not terminate at w. This case will eventually occur after, at most, $\Delta(G)$ iterations. This proof of Vizing's theorem immediately yields a polynomial algorithm running in time $O(|E||V|\Delta(G))$ to obtain a proper edge $(\Delta(G) + 1)$-coloring of a simple graph. More efficient algorithms have been designed and used in practical instances.

This edge-coloring procedure can be applied to a complete graph, in particular to K_{2n-1} to obtain an edge coloring with $2n - 1$ colors, from which an edge coloring of K_{2n} with the same number of colors can be obtained. It will provide a 1-factorization of graph K_{2n}.

Remark 5. Given any arbitrary 1-factorization of K_{2n}, there exists an order for coloring its edges by the procedure described above that will produce exactly this 1-factorization. Contrarily to the circle method, this procedure may generate any 1-factorization of K_{2n}.

16. Concluding remarks

We have realized how adequate graph models are for representing round-robin tournaments. Each game involves two teams and is an invitation to represent it by an edge linking two nodes. Complete graphs also appear very clearly whenever every two teams have to play a game against each other. Since a team cannot play more than one game in a round, we come across 1-factorizations to represent timetables.

Assigning orientations to the edges (which is a fundamental idea in graph theory) of the graph illustrates visually the fact that we have, in some circumstances, home games and away games. Having chosen to use graph-theoretical models for sports scheduling problems, we are in a situation where various tools of graph theory can be applied not only for representing a schedule but also to construct it.

Graphs help to provide a deeper understanding of the techniques used for exploring the set of 1-factorizations or, more generally, that of edge colorings.

Acknowledgments

The authors are grateful to Rafael A. Melo for revising a preliminary version of this manuscript. Work of Celso C. Ribeiro was supported by research grants CNPq 309869/2020-0 and FAPERJ E-26/200.926/2021.

References

Akiyama, J., Kano, M., 1985. Factors and factorizations of graphs - a survey. *Journal of Graph Theory* 9, 1694–1710.

Anderson, I., 1997. *Combinatorial Designs and Tournaments*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, New York.

Anderson, I., 2006. *Combinatorial Designs and Tournaments* (2nd edn.). Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, New York.

Berge, C., 1985. *Graphs and Hypergraphs*. Elsevier, Amsterdam.

Briskorn, D., 2008. Feasibility of home-away-pattern sets for round robin tournaments. *Operations Research Letters* 36, 283–284.

Caccetta, L., Mardiyono, S., 1992. Premature sets of one-factors. *Australasian Journal of Combinatorics* 5, 229–252.

Cao, Y., Wu, W., Yagiura, M., 2022. A metaheuristic algorithm for the carry-over effect value minimization problem in round-robin tournaments. *Journal of Advanced Mechanical Design, Systems, and Manufacturing* 16, 4.

Diestel, R., 2010. *Graph Theory* (4th edn.). Graduate Texts in Mathematics, Vol. 173. Springer, Berlin.

Dinitz, J.H., Stinson, D.R., 2005. On assigning referees to tournament schedules. *Bulletin of the Institute of Combinatorics and its Applications* 44, 22–28.

Erzurumluoglu, A., 2018. Constructive day-balanced round-robin tournaments with partitions. *Discrete Applied Mathematics* 235, 81–91.

Fiorini, S., Wilson, R.J., 1977. *Edge-colourings of graphs*. Pitman, London.

Folkman, J., Fulkerson, D.R., 1969. Edge-colorings in bipartite graphs. In *Combinatorial Mathematics and its Applications*. University of North Carolina Press, Chapel Hill, NC, pp. 561–578.

Gabow, H.N., Nishizeki, T., Kariv, D.O.L., Terada, O., 1985. Algorithms for edge-coloring graphs. Technical Report TRECIS-8501, Tohoku University.

Geinoz, A., Ekim, T., de Werra, D., 2008. Construction of balanced sports schedules using partitions into subleagues. *Operations Research Letters* 36, 279–282.

Goossens, D.R., Spieksma, F.C.R., 2012. Soccer schedules in Europe: an overview. *Journal of Scheduling* 15, 641–651.

Gould, R.J., 1988. *Graph theory*. Benjamin Cummings, San Francisco, CA.

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

Ikebe, Y., Tamura, A., 2008. On the existence of sports schedules with multiple venues. *Discrete Applied Mathematics* 156, 1694–1710.

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

Kirkman, T.P., 1847. On a problem in combinations. *The Cambridge and Dublin Mathematical Journal* 2, 191–204.

Kirkman, T.P., 1850. Note on an unanswered prize question. *The Cambridge and Dublin Mathematical Journal* 5, 255–262.

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

Kotzig, A., 1972. Decompositions of complete graphs into regular bichromatic factors. *Discrete Mathematics* 2, 383–387.

Lambrechts, E., Ficker, A.M.C., Goossens, D.R., Spieksma, F.C.R., 2018. Round-robin tournaments generated by the circle method have maximum carry-over. *Mathematical Programming* 172, 277–302.

Mendelsohn, E., Rodney, P., 1994. The existence of court balanced tournament designs. *Discrete Mathematics* 133, 207–216.

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

Mullin, R.C., Wallis, W.D., 1975. The existence of Room squares. *Aequationes Mathematicae* 13, 1–7.

Post, G., Woeginger, G.J., 2006. Sports tournaments, home-away assignments, and the break minimization problem. *Discrete Optimization* 3, 165–173.

Rasmussen, R.V., Trick, M.A., 2008. Round robin scheduling – a survey. *European Journal of Operational Research* 188, 617–636.

Rees, R., Wallis, W.D., 1991. The spectrum of maximal sets of one-factors. *Discrete Mathematics* 97, 357–369.

Ribeiro, C.C., 2012. Sports scheduling: problems and applications. *International Transactions in Operational Research* 19, 201–226.

Ribeiro, C.C., Urrutia, S.A., de Werra, D., 2023. *Combinatorial Models for Scheduling Sports Tournaments*. Springer, Berlin.

Room, T.G., 1955. A new type of magic square. *Mathematical Gazette* 39, 307–307.

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

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

Schellenberg, P., van Rees, G., Vanstone, S., 1977. The existence of balanced tournament designs. *Ars Combinatoria* 3, 303–318.

Suzuka, A., Miyashiro, R., Yoshise, A., Matsui, T., 2007. The home-away assignment problems and break minimization/maximization problems in sports scheduling. *Pacific Journal of Optimization* 3, 113–133.

Tanaka, T., Miyashiro, R., 2016. Fair home-away tables in sports scheduling. *Journal of Advanced Mechanical Design, Systems, and Manufacturing* 10, 3, JAMDSM0037.

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

Vizing, V.G., 1964. On an estimated of the chromatic class of a p-graph. *Diskrete Analiz.* 3, 25–30. in Russian.

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

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., 1982. Minimizing irregularities in sports schedules using graph theory. *Discrete Applied Mathematics* 4, 217–226.

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