Fair home–away tables in sports scheduling

Takahiro TANAKA* and Ryuhei MIYASHIRO**

* Graduate School of Engineering, Tokyo University of Agriculture and Technology
2-24-16 Naka-cho, Koganei-shi, Tokyo 184-8488, Japan
** Institute of Engineering, Tokyo University of Agriculture and Technology
2-24-16 Naka-cho, Koganei-shi, Tokyo 184-8488 Japan
E-mail: r-miya@cc.tuat.ac.jp

Received 15 November 2015

Abstract
This paper considers fair home–away tables in sports scheduling. A home–away table defines where each match is held in a round-robin tournament, and the quality of the tournament schedule strongly depends on the home–away table. Although some home–away tables considering fairness have been proposed, some unfair aspects remain in them. We propose three kinds of home–away tables, an even-breaks table, a consecutive-breaks table, and a $k$-interval table, which are fairer than the previously proposed home–away tables. To find consecutive-breaks and $k$-interval home–away tables consistent with a given timetable, we propose algorithms based on 2-satisfiability. We also introduce a necessary condition on the number of teams where consecutive-breaks and $k$-interval home–away tables exist.

Key words: Sports scheduling, Round-robin tournament, Home–away table, Timetable, Schedule, 2-satisfiability

1. Introduction

Sports scheduling is a research area that deals with the scheduling of sports competitions. Many sports tournaments employ a round-robin format, in which each team competes against all other teams. In some professional sports, each team has a home stadium and a match is usually held at the home stadium of one of the two competing teams. The team playing at its home stadium is typically considered to have an advantage over the visiting opponent. Accordingly, the scheduling of home and away matches is an important task.

A schedule of a single round-robin tournament is represented as a pair comprising a timetable and a HAT, explained as follows.

2. Home–away table
2.1. Preliminaries

In this paper, a single round-robin tournament that satisfies the following conditions is considered:

- The number of teams is $2n$ ($n \in \mathbb{N}$).
- The number of slots (i.e., the rounds of matches) is $2n - 1$.
- Each team plays one match in each slot.
- Each team plays every other team once in a tournament.
- Each team has its own home stadium, and each match is held at the home of one of the two competing teams.

When a team plays a match at the home of the opponent, the match is said to be “played away” for the team.

A schedule of a single round-robin tournament is represented as a pair comprising a timetable and a HAT, explained as follows.
A timetable is a two-dimensional table whose rows are indexed by a set of teams $T = \{1, 2, \ldots, 2n\}$ and columns are indexed by a set of slots $S = \{1, 2, \ldots, 2n - 1\}$, and whose $(t, s)$ element, denoted by $\tau(t, s)$, shows the opponent of team $t$ in slot $s$. A timetable should meet the following conditions to complete the schedule of a single round-robin tournament:

- The $r^{th}$ row of a timetable is a permutation of $T \setminus \{t\}$.
- For any $(t, s) \in T \times S$, $\tau(t, s), s = t$.

Throughout this paper, we assume that a timetable always satisfies these conditions.

A home–away table (HAT) is a two-dimensional table whose rows are indexed by $T$ and columns are indexed by $S$, and whose $(t, s)$ element is “H” or “A” for each $(t, s) \in T \times S$, where “H” and “A” mean that team $t$ plays at home and plays away in slot $s$, respectively.

For a given timetable, a HAT is consistent with the timetable when the following holds for any $(t, s) \in T \times S$: if the $(t, s)$ element of a HAT is “H”, then its $\tau(t, s), s$ element is “H”, otherwise “A.”

A HAT is said to be feasible if there exists at least one timetable with which the HAT is consistent; otherwise, the HAT is said to be infeasible. Conversely, a timetable is said to be corresponding to a HAT if the HAT is consistent with the timetable.

A schedule is a pair comprising a timetable and a HAT consistent with the timetable. Figure 1 shows an example of a schedule of six teams.

### 2.2. Breaks

To construct a schedule of a round-robin tournament, there are three approaches:

- Make the schedule all at once; that is, construct a timetable and a consistent HAT simultaneously.
- Fix a HAT first and then construct a timetable corresponding to the HAT.
- Fix a timetable first and then construct a HAT consistent with the timetable.

In this paper, we consider the last approach; in addition, we assume that a timetable is given in advance. Thus, our task is to construct an appropriate HAT consistent with a given timetable.

If there is no constraint on a HAT, then constructing a HAT consistent with a given timetable is trivial; e.g., one could assign “H” to each $(t, s) \in T \times S$ element such that $t < \tau(t, s)$, and assign “A” otherwise. However, such a HAT is not necessarily desirable. The quality of a HAT is often evaluated by the number of breaks, defined as follows.

When a team plays at home (“H”) in both slots $s - 1$ and $s (s \in S \setminus \{1\})$ in a HAT, it is said that the team has an $HH$-break in slot $s$; when a team plays away (“A”) in both slots $s - 1$ and $s$ in a HAT ($s \in S \setminus \{1\}$), it is said that the team has an $AA$-break in slot $s$. $HH$-breaks and $AA$-breaks are collectively referred to as breaks. The number of breaks in a HAT is the sum of the number of breaks among all teams.

Figure 2 is a HAT with four breaks. In this paper, a break is represented as a line under a corresponding element in a HAT (“H” and “A”) as shown in Figs. 1 and 2.
2.3. Previous research

In scheduling of round-robin tournaments, a break is often considered as an undesirable event. The number of breaks in a HAT usually must be kept small. The following theorem gives a lower bound on the number of breaks.

**Theorem 2.1** (de Werra 1980)

A feasible HAT for $2n$ teams has at least $2n - 2$ breaks. In addition, for any $n \geq 1$, there exists a feasible HAT for $2n$ teams with $2n - 2$ breaks.

However, not all timetables have a corresponding HAT with $2n - 2$ breaks. Accordingly, the following problem requires our attention.

**Problem 2.2**

Input: A timetable for $2n$ teams.
Output: A HAT with $2n - 2$ breaks that is consistent with the timetable if such a HAT exists, otherwise “none.”

**Corollary 2.3** (Miyashiro and Matsui 2005)

Problem 2.2 is solvable in $O(n^3)$.

Although a HAT with $2n - 2$ breaks is optimal in terms of minimizing the number of breaks, in such a HAT, $2n - 2$ teams have one break while other two teams have no break (Fig. 2). This is somewhat unfair, and so a HAT such that each team has exactly one break is desirable. Such a HAT is called an equitable HAT. Figure 3 shows an example of an equitable HAT. In addition to Problem 2.2, the following problem interests us.

**Problem 2.4**

Input: A timetable for $2n$ teams.
Output: An equitable HAT that is consistent with the timetable if such a HAT exists, otherwise “none.”

**Corollary 2.5** (Miyashiro and Matsui 2005)

Problem 2.4 is solvable in $O(n^3)$.

3. Consecutive-breaks HAT

3.1. Definition

In the previous section, the problem of finding a HAT with the minimum number, $2n - 2$, of breaks, and the problem of finding an equitable HAT were mentioned. An equitable HAT is slightly fairer than a HAT with $2n - 2$ breaks.

However, an equitable HAT still contains some unfairness. In an equitable HAT consistent with a timetable for $2n$ teams, $n$ teams have one AA-break, whereas the other $n$ teams have one HH-break (Fig. 3). This follows from the next theorem.

**Theorem 3.1** (Régin 2001)

In each slot in a feasible HAT, the number of HH-breaks and the number of AA-breaks are equal. Accordingly, in a feasible HAT the number of HH-breaks and the number of AA-breaks are equal.

**Proof**

In a feasible HAT, both the number of ‘H’s and that of ‘A’s should be $n$ in each slot. Let the number of HH-breaks in slot $s$ ($s \in \{2, 3, \ldots, 2n - 1\}$) be $h_s$. Since there are $n$ teams that have ‘H’ in slot $s$, there are exactly $n - h_s$ teams that have ‘A’ in slot $s - 1$ and ‘H’ in slot $s$. Consequently, there are $h_s$ teams that have ‘A’s in both slots $s - 1$ and $s$, i.e., an...
Slots

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | A | A | H | A | H | H | A |
| 2 | A | H | H | A | A | H | A |
| 3 | A | H | A | A | H | H | A |
| 4 | A | H | A | H | A | A | H |
| 5 | H | H | A | H | A | 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 |

Fig. 4 Even HAT.

AA-break, because there are $n$ teams that have ‘A’ in slot $s - 1$. This completes the proof of the first statement of this theorem. The second statement immediately follows from the first one.

In many sports competitions, an away match is harder for a team than a home match. Accordingly, an AA-break is usually a difficulty for a team but an HH-break is not. We introduce an even-breaks HAT (even HAT), which is fairer than an equitable HAT in this regard.

**Definition 3.2**

*An even-breaks HAT (even HAT) is a HAT such that each team has exactly one HH-break and one AA-break.*

Figure 4 shows an example of an even HAT.

We next define a consecutive-breaks HAT (consecutive HAT) as a special case of an even HAT.

**Definition 3.3**

*In an even HAT, an interval between breaks for a team is defined as $s' - s$, where $s$ and $s'$ are respectively the slots in which the team has its first and second breaks.*

**Definition 3.4**

*A consecutive-breaks HAT (consecutive HAT) is defined as an even HAT such that the interval between breaks for all teams is two.*

Suppose that a team in an even HAT has a larger interval between breaks than other teams have. If the team has an AA-break first, the team would be expected to play a larger number of matches with low vitality caused by the preceding AA-break; conversely, if the team has an HH-break first, the team can derive advantage from the HH-break for a longer period. To prevent that, in a consecutive HAT, an HH-break and an AA-break are placed adjacent so that their respective advantage and disadvantage are immediately canceled out. Moreover, all teams have the same interval between breaks. In this sense, a consecutive HAT is better than an even (but not consecutive) HAT. The left part of Fig. 5 shows an example of a consecutive HAT for eight teams.

We define the following problem regarding a consecutive HAT; in the next subsection, we propose an $O(n^8)$ algorithm to solve Problem 3.5.

**Problem 3.5**

*Input: A timetable of $2n$ teams.*

*Output: A consecutive HAT that is consistent with the timetable if such a HAT exists, otherwise “none.”*

**3.2. Algorithm**

In this subsection, we explain our algorithm to solve Problem 3.5.

Up to this point we have shown HA expression of a HAT using “H” and “A.” We now mention another way to express a HAT, called 0-1 expression, which was introduced in (Miyashiro, Iwasaki and Matsui 2003). By the following procedure, 0-1 expression of a HAT is obtained from its HA expression: Convert each “A” in slots 1, 3, …, $2n - 1$ and each “H” in slots 2, 4, …, $2n - 2$ in HA expression of a HAT into “0,” and convert each “H” in slots 1, 3, …, $2n - 1$ and each “A” in slots 2, 4, …, $2n - 2$ into “1.”

Figure 5 shows an example of a consecutive HAT in HA expression and 0-1 expression. In the remainder of this paper, we consider a HAT in 0-1 expression. The next corollary immediately follows from the transformation procedure between HA expression and 0-1 expression of a HAT.
Corollary 3.6

A HAT in 0-1 expression is a consecutive HAT if and only if both of the following conditions are satisfied:

- Each team having “0” in slot 1 has 2n − 3 “0”s, including “0” in slot 2n − 1, and has two “1”s in adjacent slots.
- Each team having “1” in slot 1 has 2n − 3 “1”s, including “1” in slot 2n − 1, and has two “0”s in adjacent slots.

Next, we see that the following easy problem can be formulated as a 2-satisfiability (2SAT) problem. Note that a 2SAT problem with p literals and q clauses is solvable in $O(p + q)$ (Aspval, Plass and Tarjan 1979).

Problem 3.7

Input: A timetable of 2n teams.
Output: A HAT consistent with the timetable.

Let $x_{t,s} \in \{0, 1\}$ ($t \in T, s \in S$) be a Boolean variable where 0 and 1 correspond to FALSE and TRUE, respectively.

Formulation 3.8

Find $x_{t,s} \in \{0, 1\}$ ($t \in T \times S$)
subject to $x_{t,s} \neq x_{t',s'}$ ($t', s' \in T \times S$).

To make 0-1 expression of a HAT from a solution of Formulation 3.8, fix “0” in slot s of team t if $x_{t,s} = 0$, otherwise “1.” Then, we obtain 0-1 expression of a HAT consistent with a given timetable because Constraint (1) guarantees the consistency. Hence, Formulation 3.8 is correct for Problem 3.7.

We now know Problem 3.7 can be formulated as a 2SAT problem. Our goal is to solve Problem 3.5, which can be regarded as Problem 3.7 with additional constraints. We also use 2SAT to solve Problem 3.5; however, the procedure is more complex than the one for Problem 3.7. In the following, we divide Problem 3.5 into several subproblems, and formulate each of them as a 2SAT problem.

The following definition and theorem play an important role in our algorithm.

Definition 3.9 (Miyashiro, Iwasaki and Matsui 2003)

In a HAT, team t is said to be a complement to team t’ if one of the following conditions is satisfied in each slot:

- Team t has “0” and team t’ has “1.”
- Team t has “1” and team t’ has “0.”

For example, teams $t$ and $t + 4$ ($t = 1, 2, 3, 4$) are complements to each other in Fig. 5.

Theorem 3.10

A feasible consecutive HAT for 2n teams includes n pairs of complement teams.

Proof

Let a team having its first break in the earliest slot in a consecutive HAT be $t_1$, and denote the slot by $s_1$. Note that, in a feasible HAT, no teams have the same pattern of “0”s and “1”s (or “H”s and “A”s), because such teams cannot play each other. From this fact and Theorem 3.1, there is exactly one team except team $t_1$ such that the team has its first break in slot $s_1$. Let the team be $t'_1$. Because of Definition 3.4, both teams $t_1$ and $t'_1$ have breaks in only slots $s_1$ and $s_1 + 2$, and hence they are complements to each other.

Next, continue the same procedure for teams except $t_1$ and $t'_1$. Let a team except for $t_1$ and $t'_1$ and having its first break in the earliest slot in the HAT be $t_2$, and denote the slot by $s_2$. From Theorem 3.1, there is exactly one team except team $t_2$
such that the team has its first break in slot $s_2$. Let the team be $t'_2$. The same discussion holds as in the case of teams $t_1$ and $t'_1$, so we can find that teams $t_2$ and $t'_2$ are complements to each other.

Applying the same procedure recursively, we can show that a feasible consecutive HAT for $2n$ teams includes $n$ pairs of complement teams. □

Denote an instance of Problem 3.5 by $P$. Without loss of generality, we may add the constraint to $P$ such that team 1 has “0” (“A”) in slot 1. We assume this constraint hereinafter. We define a subproblem corresponding to $P$ as follows.

**Problem 3.11**

Input: Team $t$ ($t \in T \setminus \{1\}$), slot $s$ ($2 \leq s \leq 2n - 3$), and input of $P$ (i.e., a timetable of $2n$ teams).

Output: A consecutive HAT that satisfies the following constraints if it exists, otherwise “none.”

- The HAT is consistent with the timetable.
- Team 1 has “0” in slot 1.
- Team $t$ is a complement to team 1.
- Team 1 has breaks in slots $s$ and $s + 2$.

Denote an instance of Problem 3.11 by $P'(t, s)$. From Theorem 3.10 and the definition of Problem 3.11, the following holds:

**Corollary 3.12**

Problem $P$ is feasible if and only if at least one of $P'(t, s)$ ($t \in T \setminus \{1\}$, $2 \leq s \leq 2n - 3$) is feasible.

Hence, we consider an algorithm to solve $P'(t, s)$ below.

In $P'(t, s)$, team 1 has “1”s in slots $s$ and $s + 1$, and “0”s in the other slots; team $t$ has “0”s in slots $s$ and $s + 1$, and “1”s in the other slots. In addition, in $P'(t, s)$, several elements in a HAT can be fixed because of consistency with a given timetable. Figure 6 shows an example. Because of teams 1 and $t$, each of the other teams has two elements fixed. At this point, there are four cases with respect to the fixed elements, as follows. Let a slot when a team plays team 1 be $s_1$, and a slot when the team plays team $t$ be $s_2$.

- $s_1 \notin \{s, s + 1\}$ and $s_2 \notin \{s, s + 1\}$: This team has one “0” and one “1” as fixed elements.
- $s_1 \in \{s, s + 1\}$ and $s_2 \notin \{s, s + 1\}$: This team has one “0” and one “1” as fixed elements.
- $s_1 \notin \{s, s + 1\}$ and $s_2 \in \{s, s + 1\}$: This team has two “0”s as fixed elements.
- $s_1 \in \{s, s + 1\}$ and $s_2 \notin \{s, s + 1\}$: This team has two “1”s as fixed elements.

According to the above analysis, at most four teams have two “0”s or two “1”s as fixed elements. In the following, we explain our algorithm when the number of such teams is four for the sake of simplicity, and we denote such teams by $t_1$, $t_2$, $t_3$, and $t_4$. If the number of such teams is less than four, we may additionally choose an arbitrary team(s).

We define another problem $P''(t, s_1, s_2, s_3, e_{t_1}, e_{t_2}, e_{t_3}, e_{t_4})$, where each of $e_{t_1}$, $e_{t_2}$, $e_{t_3}$, and $e_{t_4}$ is “0” or “1.” This is a subproblem of Problem $P'(t, s)$.

**Problem 3.13**

Input: Team $t$ ($t \in T \setminus \{1\}$), slot $s$ ($2 \leq s \leq 2n - 3$), and input of $P$ (i.e., a timetable of $2n$ teams); slots $s_1, s_2, s_3, s_4$ ($s_1, s_2, s_3, s_4 \in S \setminus \{1, s, 2n - 2, 2n - 1\}$) and elements $e_{t_1}, e_{t_2}, e_{t_3}, e_{t_4} (e_{t_1}, e_{t_2}, e_{t_3}, e_{t_4} \in \{0, 1\})$.

Output: A consecutive HAT that satisfies the following constraints if it exists, otherwise “none.”

- The HAT is consistent with the timetable.
- Team 1 has “0” in slot 1.
Corollary 3.14

Problem \( P'(t, s) \) \((t \in T \setminus \{1\}, 2 \leq s \leq 2n - 3)\) is feasible if and only if at least one of \( P''(t, s; s_1, s_2, s_3, s_4, e_1, e_2, e_3, e_4) \) \((s_1, s_2, s_3, s_4 \in S \setminus \{1, s, 2n - 2, 2n - 1\}, e_1, e_2, e_3, e_4 \in \{0, 1\})\) is feasible.

Note that the six constraints in Problem 3.13 can be formulated as 2SAT constraints. The remaining task is to express as 2SAT constraints the condition that a resultant HAT should be a consecutive HAT.

In Problem 3.13, elements corresponding to six teams (teams 1, 2, 2, 3, 4, and 5) can be completely fixed in a HAT. Accordingly, for each of the other teams, six elements are already fixed as one of the following cases:

(a) A team has three “0”s and three “1”s as fixed elements.
(b) A team has four “0”s and two “1”s as fixed elements, and these “1”s are placed in non-adjacent slots.
(c) A team has four “1”s and two “0”s as fixed elements, and these “0”s are placed in non-adjacent slots.
(d) A team has four “0”s and two “1”s as fixed elements, and these “1”s are placed in adjacent slots.
(e) A team has four “1”s and two “0”s as fixed elements, and these “0”s are placed in adjacent slots.
(f) A team has one “0” and five “1”s as fixed elements.
(g) A team has one “1” and five “0”s as fixed elements.

For cases (a), (b), and (c), this problem is infeasible because we consider a consecutive HAT (see Corollary 3.6). For cases (d) and (e), we can fix all the remaining elements of the team. Thus, we can add the corresponding 2SAT constraints to Formulation 3.8. For cases (f) and (g), we can fix the remaining elements except for the two slots adjacent to the one slot with “0” or “1;” the elements for these two slots should be different, and such a constraint can be easily added to 2SAT Formulation 3.8.

Consequently, Problem 3.13 can be formulated as a 2SAT problem (or it can be found to be infeasible in preprocessing). For an instance of Problem 3.5, the number of instances of Problem 3.13 is \( O(n^2) \), and each instance of Problem 3.13 is solvable in \( O(n^8) \). Hence, we have an \( O(n^8) \) algorithm to solve Problem 3.5.

Corollary 3.15

Problem 3.5 is solvable in \( O(n^8) \).

3.3. Necessary condition

In the previous subsection, we proposed an \( O(n^8) \) algorithm to solve Problem 3.5. A consecutive HAT is desirable in terms of fairness; however, unfortunately, the following theorem holds.

Theorem 3.16

A feasible consecutive HAT for 2n teams exists only if 8 \( \leq 2n \leq 10 \).

In the rest of this subsection, we explain why a consecutive HAT is infeasible for 2n \( \leq 6 \) or 2n \( \geq 12 \).

Proof of Theorem 3.16

As in the previous subsection, we use 0-1 expression of a consecutive HAT here. Let \( \beta_1(t) \) and \( \beta_2(t) \) be the slot where team \( t \) has its first and second breaks, respectively. When considering feasibility, without loss of generality we may assume that \( \beta_1(1) < \beta_1(2) < \cdots < \beta_1(n) \) and teams 1, 2, \ldots, n have “0” in slot 1; in this subsection, we make these assumptions. Note that each of these teams has two “1”s in adjacent slots between 2 and 2n - 2, and has “0”s in the other slots (Corollary 3.6).

First, we prove that no consecutive HAT exists when 2n \( \leq 6 \). The following holds for teams 1, 2, \ldots, n:

- There is no break in slot 1 (i.e., \( \beta_1(1) \geq 2 \)).
- For every team in a consecutive HAT, the interval between the two breaks is two, i.e., \( \beta_1(t) + 2 = \beta_2(t) \) \((t = 1, 2, \ldots, 2n)\).

The inequalities \( \beta_1(1) < \beta_1(2) < \cdots < \beta_1(n) \) imply that \( \beta_1(n) \geq \beta_1(1) + n - 1 \), and thus, we have \( \beta_1(n) \geq n + 1 \) and \( \beta_2(n) \geq n + 3 \) from the above relationship. On the other hand, obviously \( \beta_2(n) \leq 2n - 1 \). Consequently, \( n + 3 \leq \beta_2(n) \leq 2n - 1 \), which leads to \( 2n \geq 8 \).
Tanaka and Miyashiro, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol.10, No.3 (2016)

Next, we show that a consecutive HAT is infeasible when $2n \geq 12$, using the necessary condition for a feasible HAT proposed in (Miyashiro, Iwasaki and Matsui 2003). Figure 7 shows a part of a consecutive HAT for 12 teams. For these six teams, the row indexed by “#0” shows the total number of “0”s in each slot, the row indexed by “#1” shows that the number of “1”s, and the row indexed by “min” shows the minimum of these two numbers. In each slot, the number of possible matches among the six teams can be bounded by the numbers in the row “min.” The reason for this is that every match needs a pair of “0” and “1.” The sum total of the numbers in the row “min” is 12 in Fig. 7. However, to complete a single round-robin tournament, there must be $6 \times 5/2 = 15$ matches among these six teams. Hence, the consecutive HAT in Fig. 7 is infeasible.

Moreover, for teams 1, 2, . . . , $n$ in any consecutive HAT for 12 teams, the values in the row “#1” are at most two, and thus, the values in the row “min” are equal to those in “#1.” Therefore, the sum of the numbers in the row “min” is always 12, implying infeasibility.

In general, among teams 1, 2, . . . , $n$ in any consecutive HAT for $2n$ teams, the sum of the values in “min” is $2n$ and the number of matches to be held is $n(n - 1)/2$. Hence, $2n \geq n(n - 1)/2$ must hold for a consecutive HAT to be feasible, which leads to $2n \leq 10$.

Since a feasible consecutive HAT can exist only when $2n = 8$ and 10, we should discuss the computational complexity of the proposed $O(n^8)$ algorithm for finding a consecutive HAT consistent with a given timetable. If we regard $n$ as an input size, the proposed algorithm is a polynomial-time algorithm; however, a feasible consecutive HAT can exist when $8 \leq 2n \leq 10$. Hence, $n^8$ can be regarded as a constant. In this sense, the number of possible HATs is $2^{2n(2n-1)/2}$, which can also be regarded as a (huge) constant. Accordingly, an exhaustive algorithm and the proposed algorithm have the same complexity, $O(1)$. However, it should be noted that $O(n^8)$ is much better than $\Omega(2^{2n(2n-1)/2})$, even though we did not evaluate the constant factor of these two algorithms.

4. $k$-interval HAT

As described in the previous section, consideration of a consecutive HAT is insufficient because many round-robin tournaments are held with more than 10 teams. Accordingly, we extend the concept of a consecutive HAT as follows.

**Definition 4.1**

Let $k$ be a positive even integer. A $k$-interval HAT is an even HAT such that the interval of breaks for any team is exactly $k$.

In a $k$-interval HAT, the respective advantage and disadvantage caused by an HH-break and an AA-break are not immediately canceled out if $k \geq 4$, but all teams have the same interval between breaks. In this sense, a $k$-interval HAT is fairer than an even HAT.

A consecutive HAT is a 2-interval HAT. In the same way as in the proof for a consecutive HAT, it is not difficult to see that a $k$-interval HAT is infeasible when $2n$ is less than $2k + 4$ or greater than $4k + 2$.

**Theorem 4.2**

A feasible $k$-interval HAT for $2n$ teams exists only if $2k + 4 \leq 2n \leq 4k + 2$.

For a $k$-interval HAT, we consider the following problem.
Table 1 Complexity and lower and upper bounds on the number of teams for \( k \)-interval HATs.

| \( k \) | Complexity | LB | UB |
|--------|------------|----|----|
| 2      | \( O(n^3) \) | 8  | 10 |
| 4      | \( O(n^{12}) \) | 12 | 18 |
| 6      | \( O(n^{18}) \) | 16 | 26 |
| 8      | \( O(n^{20}) \) | 20 | 34 |
| \( \cdots \) | \( \cdots \) | \( \cdots \) | \( \cdots \) |
| \( k \) | \( O(n^{2k+4}) \) | \( 2k + 4 \) | \( 4k + 2 \) |

LB: lower bound; UB: upper bound.

Table 1 shows the computational complexity to solve Problem 4.3 and the upper and lower bounds on the number of teams where a feasible \( k \)-interval HAT exists. From Table 1, we can see that \( k = 6 \) is large enough for a real round-robin tournament.

5. Experiments

In the previous section, we proposed the \( k \)-interval HAT as an expansion of the consecutive HAT, and constructed an algorithm to find a feasible \( k \)-interval HAT for a given timetable if it exists. We also found that, if there is a feasible \( k \)-interval HAT, the number of teams \( 2n \) holds \( 2k + 4 \leq 2n \leq 4k + 2 \) (Theorem 4.2). However, the relationship \( 2k + 4 \leq 2n \leq 4k + 2 \) is a necessary condition for a feasible \( k \)-interval HAT, but not a sufficient condition. For instance, Fig. 8 is an infeasible \( k \)-interval HAT for \((k, 2n) = (6, 18)\), which satisfy \( 2k + 4 \leq 2n \leq 4k + 2 \) (this infeasibility was confirmed by integer programming; see the next paragraph). Here we come to a natural question here: How many feasible \( k \)-interval HATs exist among all \( k \)-interval HATs that hold \( 2k + 4 \leq 2n \leq 4k + 2 \)? To answer this question, we perform computational experiments to check the feasibility of \( k \)-interval HATs.

For \( 8 \leq 2n \leq 24, k = 2, 4, 6, \) and up to permutation of the teams, we enumerated all \( k \)-interval HATs that satisfy \( 2k + 4 \leq 2n \leq 4k + 2 \). Then, to decide their feasibility, we formulated integer programming problems; see (Nemhauser and Trick 1998) for an integer programming formulation for checking the feasibility of a HAT. The integer programming problems were solved by using IBM ILOG CPLEX 12.6.2.0 (IBM ILOG 2015) as an integer programming solver.

**Problem 4.3**

Input: A timetable of \( 2n \) teams and a positive even integer \( k \).

Output: A \( k \)-interval HAT consistent with the timetable if such a HAT exists, otherwise “none.”

For Problem 4.3, we obtained the following result. We omit a description of the algorithm, since it is quite similar to the algorithm for a consecutive HAT.

**Corollary 4.4**

**Problem 4.3 is solvable in** \( O(n^{2k+4}) \).

Fig. 8 Infeasible 6-interval HAT of 18 teams.
Table 2 Number of feasible and infeasible $k$-interval HATs.

| $k$ | $2n$ | Feasible | Infeasible |
|-----|------|----------|------------|
| 2   | 8    | 1        | 0          |
| 2   | 10   | 6        | 0          |
| 4   | 12   | 1        | 0          |
| 4   | 14   | 8        | 0          |
| 4   | 16   | 45       | 0          |
| 4   | 18   | 220      | 0          |
| 6   | 16   | 1        | 0          |
| 6   | 18   | 4        | 6          |
| 6   | 20   | 55       | 11         |
| 6   | 22   | 358      | 6          |
| 6   | 24   | 1820     | 0          |
| 6   | 26   | 8568     | 0          |

Table 2 shows that the number of feasible and infeasible $k$-interval HATs. All 2- and 4-interval HATs are feasible, and the number of infeasible 6-interval HATs is small.

6. Conclusion

In this research, we proposed three kinds of HATs for fair scheduling of single round-robin tournaments:

- Even-breaks HAT (even HAT): A HAT such that each team has one HH-break and one AA-break.
- Consecutive-breaks HAT (consecutive HAT): An even HAT such that the interval between the breaks is exactly two for each team.
- $k$-interval HAT: An even HAT such that the interval between the breaks is exactly $k$ for each team, where $k$ is even.

We proposed an $O(n^8)$ algorithm to find a consecutive HAT for a given timetable, and an $O(n^{2k+4})$ algorithm for a $k$-interval HAT. Our algorithms divide the problems of finding desirable HATs into several subproblems, and formulate each subproblem as a 2SAT problem. In addition, we determined upper and lower bounds on the number of teams where feasible $k$-interval HATs can exist. Computational experiments showed that for $k = 2, 4, 6$, almost all $k$-interval HATs between these bounds are feasible.

For the problems we proposed, there are several areas of future work. One possible direction is to pursue a sufficient condition for a feasible $k$-interval HAT. This question corresponds to a problem of finding good characterization of feasible HATs, which is a long-standing open problem in sports scheduling (de Werra 1980, Miyashiro, Iwasaki and Matsui 2003, Briskorn 2008, Horbach 2010). Another extension is to consider a HAT such that each team has one AA-break and one HH-break whose interval is at most $k$, not exactly $k$. If $k$ is set to $2n - 4$, this problem is reduced to finding an even HAT.

References

Aspval, D., Plass, M. F. and Tarjan, R. E., A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas. Information Processing Letters, Vol. 8, No. 3 (1979), pp. 121–123.

Briskorn, D., Feasibility of Home–away-Pattern Sets for Round Robin Tournaments, Operations Research Letters, Vol. 36, No. 3 (2008), pp. 283–284.

de Werra, D., Geography, Games, and Graphs, Discrete Applied Mathematics, Vol. 2, No. 4 (1980), pp. 327–337.

Horbach, A., A Combinatorial Property of the Maximum Round Robin Tournament Problem, Operations Research Letters, Vol. 38, No. 2 (2010), pp. 121–122.

IBM ILOG, IBM ILOG CPLEX 12.6.2.0 (2015).

Miyashiro, R., Iwasaki, H. and Matsui, T., Characterizing Feasible Pattern Sets with a Minimum Number of Breaks, in Burke, E. and De Causmaecker, P., eds., Practice and Theory of Automated Timetabling IV, Lecture Notes in Computer Science, Vol. 2740 (2003), pp. 78–99, Springer, Berlin.

Miyashiro, R. and Matsui, T., A Polynomial-Time Algorithm to Find an Equitable Home–Away Assignment, Operations Research Letters, Vol. 33, No. 3 (2005), pp. 235–241.

Nemhauser, G. L. and Trick, M. A., Scheduling a Major College Basketball Conference, Operations Research, Vol. 46, No. 1 (1998), pp. 1–8.

Région, J.-C., Minimization of the Number of Breaks in Sports Scheduling Using Constraint Programming, in Freuder,
E. C. and Wallace, R. J., eds., Constraint Programming and Large Scale Discrete Optimization, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 57 (2001), pp. 115–130, American Mathematical Society, Providence.