Department of Social Systems and Management

Discussion Paper Series

No. 1144

The Home-Away Assignment Problems and Break Minimization/Maximization Problems in Sports Scheduling

Ayami SUZUKA Ryuhei MIYASHIRO
Akiko YOSHISE Tomomi MATSUI

January 12, 2006

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

The Home-Away Assignment Problems and Break Minimization/Maximization Problems in Sports Scheduling

Ayami SUZUKA[1], Ryuhei MIYASHIRO[2],
Akiko YOSHISE[3], and Tomomi MATSUI[4]

[1] Graduate School of Systems and Information Engineering, University of Tsukuba,
Tsukuba, Ibaraki 305-8573, Japan. asuzuka@sk.tsukuba.ac.jp
[2] Institute of Symbiotic Science and Technology,
Tokyo University of Agriculture and Technology,
Koganei, Tokyo 184-8588, Japan. r-miya@cc.tuat.ac.jp
[3] Graduate School of Systems and Information Engineering, University of Tsukuba,
Tsukuba, Ibaraki 305-8573, Japan. yoshise@sk.tsukuba.ac.jp
[4] Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan.
tomomi@mist.i.u-tokyo.ac.jp

Abstract. Suppose that we have a timetable of a round-robin tournament with a number of teams, and distances among their homes. The home-away assignment problem is to find a home-away assignment that minimizes the total traveling distance of the teams. This paper also deals with the break minimization (maximization) problem, which finds a home-away assignment that minimizes (maximizes) the number of breaks, i.e., the number of occurrences of consecutive matches held either both at away or both at home for a team. Part of the aim of this paper is to give a unified view to the three problems, the break minimization/maximization problems and the home-away assignment problem. We see that optimal solutions of the break minimization/maximization problems are obtained by solving the home-away assignment problem.

For the home-away assignment problem, we propose a formulation as an integer program, and some rounding algorithms. We also provide a technique to transform the home-away assignment problem to MIN RES CUT and apply Goemans and Williamson's algorithm for MAX RES CUT, which is based on a positive semidefinite programming relaxation, to the obtained MIN RES CUT instances. Computational experiments show that our approaches quickly generate solutions of good approximation ratios.

Keywords: sports scheduling; dependent randomized rounding; semidefinite programming; approximation algorithm; MIN 2SAT; MIN RES CUT; MAX RES CUT.

1 Introduction

Sports scheduling has recently become a popular topic in the area of scheduling (see Chapter 52 (Sports Scheduling) [6]). Due to the variety of goals and requirements in sports scheduling, there are many optimization problems arising from sports scheduling. Among others the home-away assignment problem and the break minimization (maximization) problem are well addressed in the studies. The *home-away assignment problem* is to assign home or away to each match of a given timetable of round-robin tournament so as to minimize the total traveling distance of the teams involved in the tournament, and the *break minimization (maximization) problem* is to find a home-away assignment and minimizes (maximizes) the number of breaks (consecutive pairs of home-games or away-games).

We refer to some previous results on the two problems. The break minimization problem has been dealt in [15, 17, 7, 13]. In [13], Miyashiro and Matsui transformed the break minimization problem to MAX RES CUT and applied Goemans and Williamson's algorithm for MAX RES CUT. We extended the technique to the home-away assignment problem in [18] by introducing a different technique to transform the home-away assignment problem to MIN RES CUT, and proposed an algorithm based on positive semidefinite programming (SDP) relaxation. In [19], the authors formulated the home-away assignment problem as an integer linear programming problem and proposed a randomized algorithm based on Bertsimas, Teo and Vohra's dependent randomized rounding method [2]. Some of the above results suggest close relations among the break minimization/maximization problems and the home-away assignment problem. In fact, the independent works [12] and [22] provide partial answers to the issue.

Part of the aim of this paper is to give a more unified view to the problems. In the paper, we show the equivalence among the break minimization problem, the break maximization problem, and the (constant case) home-away assignment problem explicitly, and provide approximation algorithms for the problems, and refinement of the proofs and the results in the previous papers [12, 13, 18, 19]. We also give comprehensive comparison of computational efficiency of our algorithms to the problems.

This paper is organized as follows. The home-away assignment problem and the break minimization/maximization problems are defined in Section 2 and Section 3, respectively. We also give an equivalence theorem (Theorem 1) of the problems in Section 3. In Section 4, we propose a formulation of the home-away assignment problem as an integer program and randomized rounding algorithms. In Section 5, we transform the problem to MIN RES CUT and propose an algo-

[1] Some of the results in this paper appear in our papers [12, 13, 18, 19].

[2] This paper is also referred to as A. Suzuka, R. Miyashiro, A. Yoshise and T. Matsui, "The Home-Away Assignment Problems and Break Minimization/Maximization Problems in Sports Scheduling," METR2006-05, Department of Mathematical Engineering and Information Physics, Faculty of Engineering, The University of Tokyo, 2006.

$T\backslash S$	1	2	3	4	5	6	7
1	8	5	2	3	4	6	7
2	6	7	1	8	3	5	4
3	7	6	8	1	2	4	5
4	5	8	7	6	1	3	2
5	4	1	6	7	8	2	3
6	2	3	5	4	7	1	8
7	3	2	4	5	6	8	1
8	1	4	3	2	5	7	6

$T\backslash S$	1	2	3	4	5	6	7
1	H	H	H	A	H	A	A
2	A	H	A	H	A	H	A
3	A	A	A	H	H	H	A
4	H	A	H	H	A	A	H
5	A	A	H	A	A	A	H
6	H	H	A	A	H	H	A
7	H	A	A	H	A	A	H
8	A	H	H	A	H	H	H

Fig. 1. A timetable and HA-assignment of eight teams

rithm based on SDP relaxation. Section 6 reports the results of computational experiments.

2 Home-Away Assignment Problem

We introduce a mathematical definition of the home-away assignment problem. Throughout this paper, we deal with a round-robin tournament with the following properties:

- the number of teams (or players etc.) is $2n$, where n is a positive integer;
- the number of *slots*, i.e., the days when matches are held, is $2n-1$;
- each team plays one match in each slot;
- each team has its home, and each match is held at the home of one of the playing two teams;
- each team plays every other team once. of every other team exactly once.

Figure 1 is a *schedule* of a round-robin tournament, which is described as a pair of a timetable and home-away assignment defined below.

We denote a set of teams by $T = \{1, 2, \ldots, 2n\}$ and a set of slots by $S = \{1, 2, \ldots, 2n-1\}$. A *timetable* $\mathcal{T}$ is a matrix whose rows and columns are indexed by the set of teams T and the set of slots S, respectively. Each entry $\tau(t,s)$ $((t,s) \in T \times S)$ of a timetable $\mathcal{T}$ shows the opponent of team t in slot s. Thus, a timetable $\mathcal{T}$ should satisfy the following conditions:

- for each team $t \in T$, the t-th row of $\mathcal{T}$ contains each element of $T \setminus \{t\}$ exactly once;
- for any $(t,s) \in T \times S$, $\tau(\tau(t,s),s) = t$.

For example, team 2 of Fig. 1 plays team 3 in slot 5, and accordingly team 3 plays team 2 in the same slot.

A team is *at home* in slot s if the team plays a match at its home in s, otherwise said to be *at away* in s. A *home-away assignment* (HA-assignment for short), say $\mathcal{A}$, is a matrix whose rows are indexed by T and columns by S. Each entry $a_{t,s}$ $((t,s) \in T \times S)$ of $\mathcal{A}$ is either 'H' or 'A,' where 'H' means that in slot s team t is at home and 'A' is at away.

Given a timetable $\mathcal{T}$, an HA-assignment $\mathcal{A}$ is said to be *consistent* with $\mathcal{T}$ if $\forall(t,s) \in T \times S$, $\{a_{t,s}, a_{\tau(t,s),s}\} = \{\mathrm{A}, \mathrm{H}\}$ holds. We say that an HA-assignment $\mathcal{A}$ is *feasible* if there exists a timetable $\mathcal{T}$ such that $\mathcal{A}$ is consistent with $\mathcal{T}$. A schedule of a round-robin tournament is described as a pair of a timetable and an HA-assignment consistent with the timetable, as Fig. 1.

A *distance matrix* $\mathcal{D}$ is a matrix with zero diagonals whose rows and columns are indexed by T such that the element $d(t,t')$ denotes the distance from the home of team t to that of team t'. In this paper, we assume that $\mathcal{D}$ is symmetric and satisfies triangle inequalities. Given a consistent pair of a timetable and an HA-assignment, the traveling distance of team t is the length of the route that starts from t's home, visits venues where matches are held in the order defined by the timetable and HA-assignment, and returns to the home after the last slot. The *total traveling distance* is the sum total of the traveling distances of all teams.

Given only a timetable of a round-robin tournament, one should decide a consistent HA-assignment to complete a schedule. In practical sports scheduling, the total traveling distance is often required to be reduced [16]. In this context, the home-away assignment problem is introduced as follows.

Home-Away Assignment Problem
Instance: a timetable $\mathcal{T}$ and distance matrix $\mathcal{D}$.
Task: find an HA-assignment that is consistent with $\mathcal{T}$ and minimizes the total traveling distance.

The complexity status of the HA-assignment problem is not yet determined, though the problem is conjectured to be NP-hard.

A set of instances satisfying that all the non-diagonal elements of $\mathcal{D}$ are 1 is called the *constant case*. In the constant case, we denote the total traveling distance with respect to $\mathcal{A}$ by $w(\mathcal{A})$. We show that the constant case is essentially equivalent to the 'break minimization/maximization problems' in the next section.

In the following, we show that in the constant case the HA-assignment problem becomes an instance of MIN 2SAT. Given a set of clauses each of which consists of at most two literals, MIN 2SAT is to find a true-false assignment to literals that minimizes the number of satisfied clauses. We introduce a propositional variable $Y_{t,s}$ for each index $(t,s) \in T \times S$ that has the value `TRUE` if and only if team t plays a match at away in slot s. Then the traveling distance of team t between slots s and $s+1$ is equal to 1 if and only if the clause $Y_{t,s} \vee Y_{t,s+1}$ has the value `TRUE`. Similarly, the traveling distance of team t before the first slot (after the last slot) is equal to 1 if and only if the variable $Y_{t,1}$ ($Y_{t,2n-1}$, respectively) has the value `TRUE`. Conversely, given a true-false assignment to variables, the corresponding HA-assignment is consistent with a given timetable $\mathcal{T} = (\tau(t,s))$ if and only if $Y_{t,s}$ is the negation of $Y_{\tau(t,s),s}$ for all $(t,s) \in T \times S$. Thus, in the constant case the home-way assignment problem becomes an instance of MIN 2SAT. Bertsimas et al. [2] proposed an algorithm for MIN kSAT and showed that the expected objective value obtained by their rounding method for MIN kSAT is at most $2(1-(1/2)^k)$ times the optimal value. In the constant case,

since the HA-assignment problem can be modeled as MIN 2SAT, the approximation ratio of the above algorithm is bounded by 3/2. For MIN 2SAT, Avidor and Zwick [1] proposed a 1.1037-approximation algorithm, which is based on SDP relaxation and sophisticated but complicated randomizing technique. In Section 3.1, we give a simple (5/4)-approximation algorithm for the constant case HA-assignment problem (see Corollary 1).

3 Break Minimization/Maximization Problems

Given an HA-assignment $\mathcal{A} = (a_{t,s})$ $((t,s) \in T \times S)$, it is said that team t has a *break* at slot s ($s \in S \setminus \{1\}$) if $a_{t,s-1} = a_{t,s} = \mathrm{A}$ or $a_{t,s-1} = a_{t,s} = \mathrm{H}$. The number of breaks in an HA-assignment is defined as the number of breaks belonging to all teams. For instance, the HA-assignment of Fig. 1 has 20 breaks, each of which is represented as a line under the corresponding entry. In practical sports scheduling, such as [14], the number of breaks in an HA-assignment is required to be reduced. The break minimization (maximization) problem is to find an HA-assignment that minimizes (maximizes) the number of breaks for a given timetable.

Break Minimization (Maximization) Problem
Instance: a timetable $\mathcal{T}$.
Task: find an HA-assignment that is consistent with $\mathcal{T}$ and minimizes (maximizes) the number of breaks.

Let the number of breaks in a home-away assignment $\mathcal{A}$ be $b(\mathcal{A})$.

Lemma 1. [22] *Let $\mathcal{A}$ be a feasible HA-assignment of $2n$ teams. Then, the following holds*: $w(\mathcal{A}) + (1/2)b(\mathcal{A}) = 2n(2n-1)$, *where $w(\mathcal{A})$ denotes the total traveling distance in the constant case.*

Proof. We denote that $\mathcal{A} = (a_{t,s})$ $((t,s) \in T \times S)$. It is easy to see that for any slot $s \in S \setminus \{1\}$, $|\{t \in T : a_{t,s-1} = a_{t,s} = \mathrm{A}\}| = |\{t \in T : a_{t,s-1} = a_{t,s} = \mathrm{H}\}|$. Thus the following holds:

$$\begin{aligned} w(\mathcal{A}) &= |\{t \in T : a(t,1) = \mathrm{A}\}| + |\{t \in T : a(t,2n-1) = \mathrm{A}\}| \\ &\quad + \textstyle\sum_{s \in S \setminus \{1\}} (2n - |\{t \in T : a_{t,s-1} = a_{t,s} = \mathrm{H}\}|) \\ &= n + n + (2n-2)2n - \textstyle\sum_{s \in S \setminus \{1\}} |\{t \in T : a_{t,s-1} = a_{t,s} = \mathrm{H}\}|) \\ &= 2n(2n-1) - (1/2) \textstyle\sum_{s \in S \setminus \{1\}} |\{t \in T : a_{t,s-1} = a_{t,s}\}|) \\ &= 2n(2n-1) - (1/2)b(\mathcal{A}). \end{aligned}$$

Λ

From the above, an HA-assignment minimizing the total traveling distance in the constant case gives an optimal solution to the break maximization problem. Miyashiro and Matsui [12] proved the following lemma, which shows that the break minimization and maximization problems are essentially equivalent. Given an HA-assignment $\mathcal{A} = (a_{t,s})$ $((t,s) \in T \times S)$, define a home-away assignment $\widetilde{\mathcal{A}} = (\widetilde{a}_{t,s})$ $((t,s) \in T \times S)$ as follows:

– for $s = 1, 3, \ldots, 2n-1$, $\widetilde{a}_{t,s} := a_{t,s}$ $(\forall t \in T)$;
– for $s = 2, 4, \ldots, 2n-2$, if $a_{t,s} = \mathrm{H}$ then $\widetilde{a}_{t,s} := \mathrm{A}$, else $\widetilde{a}_{t,s} := \mathrm{H}$ $(\forall t \in T)$.

It is obvious that when $\mathcal{A}$ is consistent with a timetable $\mathcal{T}$, $\widetilde{\mathcal{A}}$ is also consistent with $\mathcal{T}$. The definition of $\widetilde{\mathcal{A}}$ directly implies the following.

Lemma 2. [12] *Let $\mathcal{A}$ be a feasible HA-assignment of $2n$ teams. Then, the equality $b(\mathcal{A}) + b(\widetilde{\mathcal{A}}) = 4n(n-1)$ holds.*

Proof. Each team has a break at slot s $(s \in S \setminus \{1\})$ in exactly one of $\mathcal{A}$ and $\widetilde{\mathcal{A}}$. Hence, $b(\mathcal{A}) + b(\widetilde{\mathcal{A}}) = |T||S \setminus \{1\}| = 2n(2n-2) = 4n(n-1)$. Λ

From the above lemmas, we have the following.

Theorem 1. *Given a timetable $\mathcal{T}$, the following conditions of an HA-assignment $\mathcal{A}$ consistent with $\mathcal{T}$ are equivalent:*

1. *$\mathcal{A}$ minimizes the total traveling distance $w(\mathcal{A})$ in the constant case,*
2. *$\mathcal{A}$ maximizes the number of breaks $b(\mathcal{A})$,*
3. *$\widetilde{\mathcal{A}}$ minimizes the number of breaks $b(\widetilde{\mathcal{A}})$.*

The break maximization is discussed in [5, 12, 16, 22]. In the following, we describe a technique to transform the break maximization problem to UNWEIGHTED MAX RES CUT.

Let $G = (V, E)$ be an undirected graph with a vertex set V and an edge set E. For any vertex subset $V' \subseteq V$, we define $\delta(V') = \{\{v_i, v_j\} : v_i, v_j \in V,\ v_i \notin V' \ni v_j\}$. The problem UNWEIGHTED MAX RES CUT is defined as follows: given a graph $G = (V, E)$ and a set $E_{\mathrm{cut}} \subseteq \{X \subseteq V : |X| = 2\}$, find a vertex subset V' that maximizes $|\delta(V') \cap E|$ under the condition that $E_{\mathrm{cut}} \subseteq \delta(V')$ holds.

Given a timetable $\mathcal{T} = (\tau(t,s))$ $((t,s) \in T \times S)$, we construct an undirected graph $G = (V, E)$ with $V = \{v_{t,s} : (t,s) \in T \times S\}$ and $E = \{\{v_{\tau(t,s-1),s-1}, v_{t,s}\} : t \in T,\ s \in S \setminus \{1\}\}$. We also introduce $E_{\mathrm{cut}} = \{\{v_{t,s}, v_{\tau(t,s),s}\} : (t,s) \in T \times S\}$. For a feasible solution V' of this UNWEIGHTED MAX RES CUT instance, i.e., a vertex subset $V' \subseteq V$ satisfying $E_{\mathrm{cut}} \subseteq \delta(V')$, construct an HA-assignment $\mathcal{A}' = (a'_{t,s})$ $((t,s) \in T \times S)$ as follows: if $v_{t,s} \in V'$ then $a'_{t,s} = \mathrm{A}$, else $a'_{t,s} = \mathrm{H}$. Clearly, $\mathcal{A}'$ is consistent with $\mathcal{T}$. It is easy to see that for each $\mathcal{T}$ there exists a bijection between the feasible set of the UNEWEIGHTED MAX RES CUT instance and the set of consistent HA-assignments. For any $(t,s) \in T \times S \setminus \{1\}$, $a'_{t,s-1} = a'_{t,s}$ if and only if $\{v_{\tau(t,s-1),s-1}, v_{t,s}\} \in \delta(V')$. Thus we have $|\delta(V') \cap E| = b(\mathcal{A}')$, and hence the break maximization problem is formulated as UNWEIGHTED MAX RES CUT. For (UNWEIGHTED) MAX RES CUT, Goemans and Williamson [10] proposed a 0.878-approximation algorithm, and accordingly the above transformation leads a 0.878-approximation algorithm for the break maximization problem. In Section 3.1, we give a simple (3/4)-approximation algorithm for the break maximization problem (see Corollary 1).

The break minimization problem is well known in sports scheduling. Régin [15] solved instances of up to 20 teams with constraint programming. Trick [17] proposed integer programming formulations and solved instances of up to 22 teams.

Elf, Jünger and Rinaldi [7] formulated this problem as MAX CUT, and solved instances of up to 26 teams. All of those methods are based on branch-and-bound technique. Miyashiro and Matsui [13] transformed this problem to MAX RES CUT and MAX 2SAT, and proposed an algorithm based on positive semidefinite programming relaxation. It is conjectured that the break minimization problem is NP-hard [7], though the complexity status is not yet determined.

It is well-known that an HA-assignment of $2n$ teams which is consistent to a timetable has at least $2n-2$ breaks (see de Werra [4]). This lower bound and above lemmas imply the following.

Theorem 2. [4, 12, 22] *Every feasible HA-assignment $\mathcal{A}$ of $2n$ teams satisfies that $2n-2 \leq b(\mathcal{A}) \leq (2n-1)(2n-2)$ and $w(\mathcal{A}) \geq (2n-1)(n+1)$.*

Proof. If an HA-assignment has two rows that are componentwise equivalent, the assignment is inconsistent to any timetable because the corresponding teams cannot play the match between them. Thus, in any HA-assignment consistent to a timetable, at most two teams have no breaks, being at home and at away alternately. Hence, the other $2n-2$ teams have at least one break, and the number of breaks is more than or equal to $2n-2$. The lower bound $2n-2 \leq b(\widetilde{\mathcal{A}})$ gives that $b(\mathcal{A}) = 2n(2n-2) - b(\widetilde{\mathcal{A}}) \leq 2n(2n-2) - (2n-2) = (2n-1)(2n-2)$ and the inequality $w(\mathcal{A}) = 2n(2n-1) - (1/2)b(\mathcal{A}) \geq 2n(2n-1) - (1/2)(2n-1)(2n-2) = (2n-1)(n+1)$. Λ

It is known that for any even integer $2n > 0$, there exists a timetable of $2n$ teams that has a consistent HA-assignment with $2n-2$ breaks (see de Werra [4] for example). The tightness of the lower bound $2n-2 \leq b(\mathcal{A})$ implies that other inequalities described above are also tight

Elf et al. [7] reported the following results: their algorithm for the break minimization problem finds optimal HA-assignments of their instances very quickly when the given timetables of $2n$ teams had HA-assignments with $2n-2$ breaks. Miyashiro and Matsui [12] proposed an $O(n^3)$ time algorithm for deciding whether a given timetable of $2n$ teams has a consistent HA-assignment $\mathcal{A}$ satisfying $b(\mathcal{A}) = 2n-2$. Their procedure reduces an instance of those problems to $2n$ instances of 2-satisfiability problem (2SAT).

3.1 Generating an HA-assignment by Pairing Slots

Here, we propose an algorithm for generating an HA-assignment with a particular structure. A key idea of the following algorithm is similar to that of the algorithm proposed in [12].

First, we describe a procedure for generating an HA-assignment $\mathcal{A}' = (a'_{t,s})$ $((t,s) \in T \times S)$ consistent with a given timetable and satisfying $[\forall t \in T,\ \forall s \in \{1,2,\ldots,n-1\},\ a'_{t,2s-1} = a'_{t,2s}]$. For each $s \in \{1,2,\ldots,n-1\}$, assign (H, H) to $(a'_{1,2s-1}, a'_{1,2s})$, for the first step. After that, continue assigning home or away to each of other teams so as to satisfy $a'_{1,2s-1} = a'_{1,2s}$. Due to the consistency, the opponent of team 1 in slot $2s$, $\tau(1,2s)$, has to be at away in slot $2s$. So as

to satisfy $a'_{\tau(1,2s),2s-1} = a'_{\tau(1,2s),2s}$, we assign (A, A) to $(a'_{\tau(1,2s),2s-1}, a'_{\tau(1,2s),2s})$. In the same way, the opponent of team $\tau(1,2s)$ of slot $2s-1$, $\tau(\tau(1,2s),2s-1)$ has to be at home, and so as to satisfy $a'_{\tau(\tau(1,2s),2s-1)} = a'_{\tau(\tau(1,2s),2s)}$, we assign (H, H) to $(a'_{\tau(\tau(1,2s),2s-1)}, a'_{\tau(\tau(1,2s),2s)})$. Repeat this assignment procedure to the rest of teams. For the last slot $s = 2n-1$, assign home or away to each team as keeping consistency. Then it is easy to see that $\mathcal{A}'$ is consistent with a given timetable and satisfies that $[\forall t \in T,\ \forall s \in \{1,2,\ldots,n-1\},\ a'_{t,2s-1} = a'_{t,2s}]$. Similarly, we can generate an HA-assignment $\mathcal{A}'$ that is consistent with a given timetable and satisfying $[\forall t \in T,\ \forall s \in \{1,2,\ldots,n-1\},\ a'_{t,2s} = a'_{t,2s+1}]$.

Given an HA-assignment $\mathcal{A} = (a_{t,s})$ and a slot-subset $S' \subseteq S$, an HA-assignment $\mathcal{A}' = (a'_{t,s})$ obtained from $\mathcal{A}$ by *flipping slots in* S' is defined as follows:

$$a'_{t,s} = \begin{cases} a_{t,s} & (\text{if } s \notin S'), \\ \text{H} & (\text{if } s \in S' \text{ and } a_{t,s} = \text{A}), \\ \text{A} & (\text{if } s \in S' \text{ and } a_{t,s} = \text{H}). \end{cases}$$

Now we describe an algorithm for generating an HA-assignment $\mathcal{A}^*$ consistent with a given timetable.

Pairing Slots
Step 0: Execute one of Steps 1 and 2 at random.
Step 1: Generate an HA-assignment $\mathcal{A}' = (a'_{t,s})$ consistent with a given timetable and satisfying $[\forall t \in T,\ \forall s \in \{1,2,\ldots,n-1\},\ a'_{t,2s-1} = a'_{t,2s}]$. Let $\mathcal{A}^* = (a^*_{t,s})$ be an HA-assignment obtained from $\mathcal{A}'$ by flipping slots in $\{2s-1, 2s\}$ with probability 1/2 for each $s \in \{1,2,\ldots,n-1\}$ independently. Output $\mathcal{A}^*$ and stop.
Step 2: Generate an HA-assignment $\mathcal{A}' = (a'_{t,s})$ consistent with a given timetable and satisfying $[\forall t \in T,\ \forall s \in \{1,2,\ldots,n-1\},\ a'_{t,2s} = a'_{t,2s+1}]$. Let $\mathcal{A}^* = (a^*_{t,s})$ be an HA-assignment obtained from $\mathcal{A}'$ by flipping slots in $\{2s, 2s+1\}$ with probability 1/2 for each $s \in \{1,2,\ldots,n-1\}$ independently. Output $\mathcal{A}^*$ and stop.

The procedure 'Pairing Slots' gives the following.

Theorem 3. [12] *For each timetable, there exists a consistent HA-assignment* $\mathcal{A}$ *satisfying that* $b(\mathcal{A}) \geq 3n(n-1)$, $b(\widetilde{\mathcal{A}}) \leq n(n-1)$ *and* $w(\mathcal{A}) \leq (1/2)n(5n-1)$.

Proof. Assume that an HA-assignment $\mathcal{A}^*$ is obtained in Step 1 of the procedure Pairing Slots. Every team has a break at each slot $s \in \{2,4,\ldots,2n-2\}$. For each pair of a team t and a slot $s \in \{3,5,\ldots,2n-1\}$, team t has a break at slot s with probability 1/2. The expected value of the number of breaks is $\mathrm{E}[b(\mathcal{A}^*)] = 2n(n-1) + (1/2)(2n)(n-1) = 3n(n-1)$. Thus there exists an HA-assignment $\mathcal{A}$ satisfying that $b(\mathcal{A}) \geq 3n(n-1)$. The other inequalities are obtained by applying the equalities in Lemmas 1 and 2. Similarly, we can show the case that an HA-assignment $\mathcal{A}^*$ is obtained in Step 2 of the procedure Pairing Slots. Λ

From the above, we obtained an approximation algorithm for the break maximization problem and the constant case HA-assignment problem.

Corollary 1. *The procedure Pairing Slots is* (i) *a* $(3/4)$*-approximation algorithm for the break maximization problem, and* (ii) *a* $(5/4)$*-approximation algorithm for the constant case HA-assignment problem.*

Proof. (i) Theorem 2 shows that every feasible HA-assignment $\mathcal{A}$ of $2n$ teams satisfies that $2(2n-1)(n-1) \geq b(\mathcal{A})$. Thus, by the proof of Theorem 3, an HA-assignment $\mathcal{A}^*$ obtained by the procedure Pairing Slots satisfies that $\mathrm{E}[b(\mathcal{A}^*)] = 3n(n-1) = (3/4)4n(n-1) \geq (3/4)2(2n-1)(n-1)$.
(ii) Theorem 2 shows that every feasible HA-assignment $\mathcal{A}$ of $2n$ teams satisfies that $(2n-1)(n+1) \leq w(\mathcal{A})$. Thus, by the proof of Theorem 3, an HA-assignment $\mathcal{A}^*$ obtained by the procedure Pairing Slots satisfies that $\mathrm{E}[w(\mathcal{A}^*)] = (1/2)n(5n-1) = (5/4)2n(n-1/5) \leq (5/4)(2n-1)(n+1)$. Λ

4 Integer Linear Programming Formulation

In this section, we formulate the HA-assignment problem as an integer programming problem [19]. In the rest of this paper, we denote the last slot by $\hat{s}$, i.e., $\hat{s} = 2n-1$. We introduce 0-1 variables $y_{t,s}$ $((t,s) \in T \times S)$ such that $y_{t,s}$ is 1 if and only if team t is at away in slot s, and continuous variables $w_{t,s}$ $((t,s) \in T \times S \setminus \{\hat{s}\})$ where $w_{t,s}$ represents the traveling distance of team t between slots s and $s+1$. Then we can formulate the HA-assignment problem as follows:

(IP)

$$\text{min.} \quad \sum_{t \in T} \left(\sum_{s \in \{1, \hat{s}\}} d(t, \tau(t,s))\, y_{t,s} + \sum_{s \in S \setminus \{\hat{s}\}} w_{t,s} \right)$$

$$\text{s. t.} \quad w_{t,s} \geq d(t',t)\, y_{t,s} + (d(t',t'') - d(t',t))\, y_{t,s+1} \quad \begin{pmatrix} \forall (t,s) \in T \times S \setminus \{\hat{s}\}, \text{ where} \\ t' = \tau(t,s) \text{ and } t'' = \tau(t,s+1)) \end{pmatrix},$$

$$w_{t,s} \geq (d(t',t'') - d(t,t''))\, y_{t,s} + d(t,t'')\, y_{t,s+1} \quad \begin{pmatrix} \forall (t,s) \in T \times S \setminus \{\hat{s}\}, \text{ where} \\ t' = \tau(t,s) \text{ and } t'' = \tau(t,s+1)) \end{pmatrix},$$

$$y_{t,s} + y_{\tau(t,s),s} = 1 \qquad (\forall (t,s) \in T \times S),$$
$$y_{t,s} \in \{0,1\} \qquad (\forall (t,s) \in T \times S),$$

where $w_{t,s}$ $((t,s) \in T \times S \setminus \{\hat{s}\})$ are continuous variables. The constraints in **IP** are explained as follows. The first and second constraints give the lower envelope of the following four points

$$(y_{t,s}, y_{t,s+1}, w_{t,s}) \in \{(0,0,0),\ (1,0,d(t',t)),\ (0,1,d(t,t'')),\ (1,1,d(t',t''))\}$$

where $t' = \tau(t,s)$ and $t'' = \tau(t,s+1)$, because the distance matrix satisfies triangle inequalities. The third constraints guarantee that every HA-assignment corresponding to a feasible solution is consistent with the given timetable.

A linear relaxation problem **LP** is a linear programming problem obtained from **IP** by substituting the 0-1 constraints for variables $y_{t,s}$ for nonnegativity constraints $y_{t,s} \geq 0$ $(\forall (t,s) \in T \times S)$. We prove the theorem showing that **LP** has an optimal solution satisfying half-integrality on variables $y_{t,s}$ $(\forall (t,s) \in T \times S)$.

Theorem 4. [19] *Suppose that a distance matrix $\mathcal{D}$ satisfies triangle inequalities. In any extreme point optimal solution of* **LP**, $y_{t,s} \in \{0, \frac{1}{2}, 1\}$ *holds for any* $(t,s) \in T \times S$.

Proof. See Appendix.

4.1 Randomized Rounding Algorithms

Here, we propose algorithms for **IP**. In our algorithms, we solve the linear relaxation problem **LP** first. If an obtained solution is 0-1 valued, we have an optimal solution of the original problem **IP**. Otherwise, we construct a feasible solution of **IP** by rounding the obtained solution. In the following, we propose three randomized rounding algorithms. We denote an optimal solution of **LP** by $(\boldsymbol{y}^*, \boldsymbol{w}^*)$.

A1: Independent Randomized Rounding

The first algorithm generates a 0-1 valued solution as follows. For each pair of teams $\{t, t'\}$, we decide the venue of the match independently of the venue of another match. Let s be the slot when t and t' play a match, i.e., $\tau(t,s) = t'$. Then we construct a solution $\boldsymbol{y}''$ of **IP** by setting the pair of variables $(y''_{t,s}, y''_{t',s})$ to $(1,0)$ or $(0,1)$ with probability $y^*_{t,s}$ and $1 - y^*_{t,s}$, respectively. The independent rounding algorithm is similar to the LP-based approximation algorithm for MAX SAT proposed by Goemans and Williamson [9].

A2: Dependent Randomized Rounding with Random HA-assignment

As we described in Section 3, **IP** becomes an instance of MIN 2SAT in the constant case. For MIN kSAT, Bertsimas et al. [2] proposes an approximation algorithm based on randomized rounding introducing dependencies in the rounding process. Our second algorithm described below is a direct application of their algorithm to a general case. First, we construct an HA-assignment $\mathcal{A}^* = (a^*_{t,s})$ consistent with a given timetable by randomly choosing one of two possible venues for each match. Next, we execute the following procedure.

Dependent Randomized Rounding
Step 0: Generate a uniform random number $U \in (0,1]$.

Step 1: Set $y''_{t,s}$ $((t,s) \in T \times S)$ as follows:

$$y''_{t,s} = \begin{cases} 1 \left(\begin{array}{l} \text{if } [y^*_{t,s} \geq U \text{ and } a^*_{t,s} \text{ is A}] \\ \text{or } [y^*_{t,s} > 1-U \text{ and } a^*_{t,s} \text{ is H}] \end{array} \right), \\ 0 \left(\begin{array}{l} \text{if } [y^*_{t,s} < U \text{ and } a^*_{t,s} \text{ is A}] \\ \text{or } [y^*_{t,s} \leq 1-U \text{ and } a^*_{t,s} \text{ is H}] \end{array} \right). \end{cases}$$

Step 2: Generate an HA-assignment $\mathcal{A}'' = (a''_{t,s})$ by assigning 'A' to $a''_{t,s}$ if $y''_{t,s} = 1$, otherwise 'H.'

It is easy to see that the above procedure outputs a feasible solution of **IP**.

A3: Dependent Randomized Rounding with the Procedure Pairing Slots

In our third algorithm, we generate an HA-assignment $\mathcal{A}^* = (a^*_{t,s})$ by an algorithm based on the procedure Pairing Slots described in Section 3.1 and execute 'Dependent Randomized Rounding' procedure described above.

A practical procedure to obtain a better solution by our randomized rounding algorithm A3 is to generate a number of initial HA-assignments $\mathcal{A}^*$ and output a solution with the best objective value. For our third algorithm, we can generate a number of initial HA-assignments $\mathcal{A}^*$ from a specific HA-assignment $\mathcal{A}'$ by randomly flipping slots in $\{2s-1, 2s\}$ at Step 1 and slots in $\{2s, 2s+1\}$ at Step 2 for each $s \in \{1, 2, \ldots, n-1\}$ several times.

4.2 Derandomization

Here we discuss the derandomization of procedures proposed in Section 4.1. We can derandomize the procedures A1 and A3 by an ordinary method of conditional probabilities (see [9] for example). In the following, we describe a derandomize procedure of A2, which extremely shortens practical computational time. Now we suppose that an optimal solution $(\boldsymbol{y}^*, \boldsymbol{w}^*)$, which is obtained by solving **LP**, satisfies half-integrality on $\boldsymbol{y}$. Then, it is easy to see that if the uniform random number U obtained at Step 0 in the procedure 'Dependent Randomized Rounding,' satisfies $0 < U \leq 1/2$, then the variables $\boldsymbol{y}''$ and the HA-assignment $\mathcal{A}''$ obtained in Steps 1 and 2 are independent of the magnitude of U. In case of $1/2 < U \leq 1$, we can also show that $\boldsymbol{y}''$ and $\mathcal{A}''$ obtained in Steps 1 and 2 are independent of the magnitude of U. Thus, we only need to execute Steps 1 and 2 only for two cases that $U \in \{\frac{1}{2}, 1\}$ and output the better solution. Clearly, a solution obtained by the above derandomized procedure satisfies that the corresponding objective function value (total traveling distance) is less than or equal to the expectation of that of solutions obtained by randomized rounding algorithm A2. In our computational experiments, we use the above derandomized procedures for randomized rounding algorithm A2.

4.3 Constant Case

In the constant case, the linear relaxation problem **LP** of **IP** has the following property.

Theorem 5. *In the constant case, the linear relaxation problem* **LP** *has a unique optimal solution* $(\boldsymbol{y}^*, \boldsymbol{w}^*)$ *satisfying* $y^*_{t,s} = 1/2$ $(\forall (t,s) \in T \times S)$ *and* $w^*_{t,s} = 1/2$ $(\forall (t,s) \in T \times S \setminus \{\hat{s}\})$.

Proof. It is clear that the solution $(\boldsymbol{y}^*, \boldsymbol{w}^*)$ defined above is feasible to **LP**. The corresponding objective value is equal to $2n(2(1/2)+(2n-2)(1/2)) = 2n^2$. First, we show that an objective value of any feasible solution $(\boldsymbol{y}, \boldsymbol{w})$ of **LP** is greater than or equal to $2n^2$. It is easy to see that the corresponding objective value Z satisfies that

$$
\begin{aligned}
Z &= \sum_{t \in T} \left(y_{t,1} + y_{t,\hat{s}} + \sum_{s \in S \setminus \{\hat{s}\}} w_{t,s} \right) \geq \sum_{t \in T} \left(y_{t,1} + y_{t,\hat{s}} + \sum_{s \in S \setminus \{\hat{s}\}} y_{t,s} \right) \\
&= (1/2) \sum_{t \in T} \left((y_{t,1} + y_{\tau(t,1),1}) + (y_{t,\hat{s}} + y_{\tau(t,\hat{s}),\hat{s}}) + \sum_{s \in S \setminus \{\hat{s}\}} (y_{t,s} + y_{\tau(t,s),s}) \right) \\
&= (1/2) \sum_{t \in T} \left(1 + 1 + \sum_{s \in S \setminus \{\hat{s}\}} 1 \right) = (1/2) 2n(1 + 1 + 2n - 2) = 2n^2.
\end{aligned}
$$

Next, we show the uniqueness. Let $(\boldsymbol{y}', \boldsymbol{w}')$ be an optimal solution of **LP**. The optimality implies that

$$
\begin{aligned}
2n^2 &= \sum_{t \in T} \left(y'_{t,1} + y'_{t,\hat{s}} + \sum_{s \in S \setminus \{\hat{s}\}} w'_{t,s} \right) \geq \sum_{t \in T} \left(y'_{t,1} + y'_{t,\hat{s}} + \sum_{s \in S \setminus \{\hat{s}\}} y'_{t,s} \right) \\
&= (1/2) \sum_{t \in T} \left((y'_{t,1} + y'_{\tau(t,1),1}) + (y'_{t,\hat{s}} + y'_{\tau(t,\hat{s}),\hat{s}}) + \sum_{s \in S \setminus \{\hat{s}\}} (y'_{t,s} + y'_{\tau(t,s),s}) \right) \\
&= (1/2) \sum_{t \in T} \left(1 + 1 + \sum_{s \in S \setminus \{\hat{s}\}} 1 \right) = (1/2) 2n(1 + 1 + 2n - 2) = 2n^2,
\end{aligned}
$$

and thus the equality $\sum_{t \in T} \sum_{s \in S \setminus \{\hat{s}\}} w'_{t,s} = \sum_{t \in T} \sum_{s \in S \setminus \{\hat{s}\}} y'_{t,s}$ holds. The feasibility of $(\boldsymbol{y}', \boldsymbol{w}')$ implies that $w'_{t,s} \geq \max\{y'_{t,s}, y'_{t,s+1}\}$ for all $(t,s) \in T \times S \setminus \{\hat{s}\}$. From the above, we have that $w'_{t,s} = y'_{t,s}$ for all $(t,s) \in T \times S \setminus \{\hat{s}\}$. Similarly,

the following inequality

$$2n^2 = \sum_{t\in T}\left(y'_{t,1} + y'_{t,\hat{s}} + \sum_{s\in S\setminus\{\hat{s}\}} w'_{t,s}\right) \geq \sum_{t\in T}\left(y'_{t,1} + y'_{t,\hat{s}} + \sum_{s\in S\setminus\{\hat{s}\}} y'_{t,s+1}\right)$$
$$= (1/2)\sum_{t\in T}\left((y'_{t,1} + y'_{\tau(t,1),1}) + (y'_{t,\hat{s}} + y'_{\tau(t,\hat{s}),\hat{s}}) + \sum_{s\in S\setminus\{\hat{s}\}} (y'_{t,s+1} + y'_{\tau(t,s+1),s+1})\right)$$
$$= (1/2)\sum_{t\in T}\left(1 + 1 + \sum_{s\in S\setminus\{\hat{s}\}} 1\right) = (1/2)2n(1+1+2n-2) = 2n^2$$

directly implies $\sum_{t\in T}\sum_{s\in S\setminus\{\hat{s}\}} w'_{t,s} = \sum_{t\in T}\sum_{s\in S\setminus\{\hat{s}\}} y'_{t,s+1}$ and thus $w'_{t,s} = y'_{t,s+1}$ for all $(t,s) \in T \times S \setminus \{\hat{s}\}$. From the above, we have the property that

$$\forall t \in T,\ \ y'_{t,1} = w'_{t,1} = y'_{t,2} = w'_{t,2} = \cdots = y'_{t,\hat{s}}.$$

If $(\boldsymbol{y}', \boldsymbol{w}') \neq (\boldsymbol{y}^*, \boldsymbol{w}^*)$, there exists an index $(t', s') \in T \times S$ satisfying $y'_{t',s'} > (1/2)$ and thus $\forall (t,s) \in T \setminus \{t'\} \times S$, $y'_{t,s} = 1 - y'_{t',s'} < (1/2)$. It contradicts the feasibility of $(\boldsymbol{y}', \boldsymbol{w}')$. Λ

From the above, we can estimate the objective values obtained by our randomized algorithms based on the linear relaxation problem **LP**. We denote the optimal value of **IP** by Z^{IP}. We also denote the objective values obtained by our first, second and third algorithms in Section 4.1 by Z^{A1}, Z^{A2} and Z^{A3}, respectively. Then the following theorem holds.

Theorem 6. *In the constant case, the followings hold:*

1. $\mathrm{E}[Z^{\mathrm{A1}}] = \mathrm{E}[Z^{\mathrm{A2}}] = n(3n-1) \leq \frac{3}{2}Z^{\mathrm{IP}}$,
2. $\mathrm{E}[Z^{\mathrm{A3}}] = (1/2)n(5n-1) \leq \frac{5}{4}Z^{\mathrm{IP}}$.

Proof. It is easy to see that an HA-assignment $\mathcal{A}_1^*$ obtained by our randomized rounding algorithm A1 satisfies that $\mathrm{E}[b(\mathcal{A}_1^*)] = (1/2)2n(2n-2) = 2n(n-1)$ and thus

$$\mathrm{E}[Z^{\mathrm{A1}}] = \mathrm{E}[w(\mathcal{A}_1^*)] = 2n(2n-1) - (1/2)\mathrm{E}[b(\mathcal{A}_1^*)] = n(3n-1).$$

Theorem 2 implies that $(2n-1)(n+1) \leq Z^{\mathrm{IP}}$ and

$$\mathrm{E}[Z^{\mathrm{A1}}] = n(3n-1) = (3/2)n(2n-2/3) \leq (3/2)(n+1)(2n-1) \leq (3/2)Z^{\mathrm{IP}}.$$

Theorem 5 directly implies that $\mathrm{E}[Z^{\mathrm{A1}}] = \mathrm{E}[Z^{\mathrm{A2}}]$.

Let $\mathcal{A}_3^*$ be an HA-assignment obtained by the procedure Pairing Slots. Theorem 5 directly implies that $\mathrm{E}[Z^{\mathrm{A3}}] = \mathrm{E}[w(\mathcal{A}_3^*)]$. The proof of Theorem 3 implies that

$$\mathrm{E}[w(\mathcal{A}_3^*)] = (1/2)n(5n-1) = (5/4)2n(n-1/5) \leq (5/4)(2n-1)(n+1).$$

From Theorem 2, it is obvious that $\mathrm{E}[Z^{\mathrm{A3}}] = \mathrm{E}[w(\mathcal{A}_3^*)] \leq (5/4)(2n-1)(n+1) \leq (5/4)Z^{\mathrm{IP}}$. Λ

The above theorem indicates that our algorithm A3 finds a solution whose objective value is better that that of A1 and A2. However, our computational experiments in Section 6 show that for a class of instances, A1 generates solutions with better approximation ratios than A2 or A3 on average.

5 Formulation as MIN RES CUT

In this section, we propose a formulation of the HA-assignment problem as MIN RES CUT and a randomized algorithm based on a positive semidefinite programming relaxation [13, 18]. First, we define the problem MIN RES CUT. Let $G = (V, E)$ be an undirected graph with a vertex set V and an edge set E. For any vertex subset $V' \subseteq V$, we define $\delta(V') = \{\{v_i, v_j\} : v_i, v_j \in V,\ v_i \notin V' \ni v_j\}$. The problem MIN RES CUT is defined as follows: given a graph $G = (V, E)$, a specified vertex $r \in V$, a weight function $\ell : E \to \mathrm{R}$, and a set $E_{\mathrm{cut}} \subseteq \{X \subseteq V : |X| = 2\}$, find a vertex subset V' that minimizes $\sum_{e \in \delta(V') \cap E} \ell(e)$ under the conditions that $r \notin V'$ and $E_{\mathrm{cut}} \subseteq \delta(V')$ hold. Here we note that the condition $r \notin V'$ is redundant for the definition of MIN RES CUT, because for any $V'' \subseteq V$, $\delta(V'') = \delta(V \setminus V'')$. The condition helps to formulate the HA-assignment problem as MIN RES CUT. It is easy to show that MIN RES CUT is NP-hard even if $\forall e \in E, \ell(e) = 1$ holds. The problem MAX RES CUT is the maximization version of MIN RES CUT, and Goemans and Williamson [10] proposed a 0.878-approximation algorithm for MAX RES CUT.

Now we formulate the HA-assignment problem as MIN RES CUT. Given a timetable $\mathcal{T} = (\tau(t, s))$ $((t, s) \in T \times S)$, let $G = (V, E)$ be an undirected graph with a vertex set V and an edge set E defined below. We introduce an artificial vertex r and define $V = \{v_{t,s} : (t, s) \in T \times S\} \cup \{r\}$, $E = \{\{v_{t,s-1}, v_{t,s}\} : t \in T, s \in S \setminus \{1\}\} \cup \{\{r, v_{t,s}\} : (t, s) \in T \times S\}$, and $E_{\mathrm{cut}} = \{\{v_{t,s}, v_{\tau(t,s),s}\} : (t, s) \in T \times S\}$. For a feasible solution V' of this MIN RES CUT instance, i.e., a vertex subset $V' \subseteq V$ satisfying $r \notin V'$ and $E_{\mathrm{cut}} \subseteq \delta(V')$, construct an HA-assignment $\mathcal{A} = (a_{t,s})$ $((t, s) \in T \times S)$ as follows: if $v_{t,s} \in V'$ then $a_{t,s} = \mathrm{A}$, else $a_{t,s} = \mathrm{H}$. This HA-assignment is consistent with $\mathcal{T}$ because each pair of vertices corresponding to a match is in $E_{\mathrm{cut}} \subseteq \delta(V')$. Obviously, for any consistent HA-assignment, there exists a unique corresponding feasible solution of the MIN RES CUT instance. Thus, for each $\mathcal{T}$, there exists a bijection between the feasible set of the MIN RES CUT instance and the set of consistent HA-assignments.

Next, we discuss the total traveling distance. In the following, we denote any singleton $\{v\}$ by v for simplicity. Given a pair of timetable $\mathcal{T}$ and an HA-assignment $\mathcal{A}$ consistent with $\mathcal{T}$, the traveling distance of team t between slots s and $s + 1$, denoted by $w_{t,s}$, is defined as follows:

$$w_{t,s} = \begin{cases} 0 & (\text{if } (a_{t,s}, a_{t,s+1}) = (\mathrm{H}, \mathrm{H})), \\ d(\tau(t, s), \tau(t, s+1)) & (\text{if } (a_{t,s}, a_{t,s+1}) = (\mathrm{A}, \mathrm{A})), \\ d(t, \tau(t, s+1)) & (\text{if } (a_{t,s}, a_{t,s+1}) = (\mathrm{H}, \mathrm{A})), \\ d(\tau(t, s), t) & (\text{if } (a_{t,s}, a_{t,s+1}) = (\mathrm{A}, \mathrm{H})). \end{cases}$$

In the following, we use the notations $t' = \tau(t,s)$, $t'' = \tau(t,s+1)$, $N(s,r) = |\{v_{t,s}, r\} \cap \delta(V')|$, $N(s+1,r) = |\{v_{t,s+1}, r\} \cap \delta(V')|$ and $N(s,s+1) = |\{v_{t,s}, v_{t,s+1}\} \cap \delta(V')|$ for simplicity. We show that the traveling distance $w_{t,s}$ satisfies the following equations:

$$\begin{aligned}
w_{t,s} &= d(t',t'')\ |v_{t,s} \cap V'|\ |v_{t,s+1} \cap V'| \\
&\quad + d(t,t'')\ (1 - |v_{t,s} \cap V'|)\ |v_{t,s+1} \cap V'| \\
&\quad + d(t',t)\ |v_{t,s} \cap V'|\ (1 - |v_{t,s+1} \cap V'|) \\
&= d(t',t'') \left\{ \frac{N(s,r) + N(s+1,r) - N(s,s+1)}{2} \right\} \\
&\quad + d(t,t'') \left\{ \frac{-N(s,r) + N(s+1,r) + N(s,s+1)}{2} \right\} \\
&\quad + d(t',t) \left\{ \frac{N(s,r) - N(s+1,r) + N(s,s+1)}{2} \right\} \\
&= \frac{d(t',t'') - d(t,t'') + d(t',t)}{2} N(s,r) \\
&\quad \frac{d(t',t'') + d(t,t'') - d(t',t)}{2} N(s+1,r) \\
&\quad \frac{-d(t',t'') + d(t,t'') + d(t',t)}{2} N(s,s+1).
\end{aligned}$$

The first equality is obvious, because $[a_{t,s} = \mathrm{A} \iff |v_{t,s} \cap V'| = 1]$ and $[a_{t,s+1} = \mathrm{A} \iff |v_{t,s+1} \cap V'| = 1]$. The second equality is obtained by applying the equations

$$|v_{t,s} \cap V'| = N(s,r), \quad |v_{t,s+1} \cap V'| = N(s+1,r), \tag{1}$$

and

$$|v_{t,s} \cap V'|\ |v_{t,s+1} \cap V'| = \frac{N(s,r) + N(s+1,r) - N(s,s+1)}{2}. \tag{2}$$

Equations (1) and (2) are obtained from the properties that $r \notin V'$ and

$$\forall V' \subseteq V,\ |\delta(V') \cap \{\{r, v_{t,s}\}, \{r, v_{t,s+1}\}, \{v_{t,s}, v_{t,s+1}\}\}| \in \{0, 2\}.$$

The third equality is trivial. Here we note that, if we employ only Equations (1), $w_{t,s}$ becomes a quadratic function of $N(s,r)$ and $N(s+1,r)$. Using Equation (2), we can transform the quadratic function to a linear function of $N(s,r)$, $N(s+1,r)$ and $N(s,s+1)$.

In a similar way, we can show that the traveling distance of team t before the first slot and after the last slot, denoted by $w_{t,0}$ and $w_{t,\hat{s}}$ respectively, satisfy that

$$\begin{aligned}
w_{t,0} &= d(t, \tau(t,1)) N(1,r), \\
w_{t,\hat{s}} &= d(\tau(t,\hat{s}), t) N(\hat{s},r).
\end{aligned}$$

From the above, the total traveling distance is represented by a linear function of variables $|e \cap \delta(V')|$ $(e \in E)$ as follows:

$$\sum_{t\in T}\sum_{s=0}^{\hat{s}} w_{t,s} = \sum_{t\in T}\sum_{s=1}^{\hat{s}-1} \left(\begin{array}{l} \dfrac{d(t',t'')-d(t,t'')+d(t',t)}{2} N(s,r) \\ + \dfrac{d(t',t'')+d(t,t'')-d(t',t)}{2} N(s+1,r) \\ + \dfrac{-d(t',t'')+d(t,t'')+d(t',t)}{2} N(s,s+1) \end{array} \right) + \sum_{t\in T} d(t,\tau(t,1))N(1,r) + \sum_{t\in T} d(\tau(t,\hat{s}),t)N(\hat{s},r).$$

Thus, by introducing an appropriate weight function $\ell : E \to \mathrm{R}_+$ (see below), the total traveling distance satisfies that

$$\sum_{t\in T}\sum_{s=0}^{\hat{s}} w_{t,s} = \textstyle\sum_{e\in E} \ell(e)|e \cap \delta(V')| = \sum_{e\in E\cap\delta(V')} \ell(e) \tag{3}$$

and the objective function value of the MIN RES CUT, with respect to $\ell(e)$, is equivalent to the total traveling distance. From the above, the HA-assignment problem is formulated as the MIN RES CUT.

We define a weight function $\ell : E \to \mathrm{R}_+$ as follows:

$$\ell(\{v_{t,s}, r\}) = \frac{d(t',t'')-d(t,t'')+d(t',t)}{2} + \frac{d(\tau(t,s-1),t')+d(t,t')-d(\tau(t,s-1),t)}{2}$$

$$(\forall t \in T, \forall s \in S \setminus \{1,\hat{s}\}),$$

$$\ell(\{v_{t,1}, r\}) = d(t,\tau(t,1)) + \frac{d(\tau(t,1),\tau(t,2))-d(t,\tau(t,2))+d(\tau(t,1),t)}{2},$$

$$\ell(\{v_{t,\hat{s}}, r\}) = d(\tau(t,\hat{s}),t) + \frac{d(\tau(t,\hat{s}-1),\tau(t,\hat{s})))}{2} + \frac{d(t,\tau(t,\hat{s}))-d(\tau(t,\hat{s}-1),t)}{2},$$

$$\ell(\{v_{t,s}, v_{t,s+1}\}) = \frac{-d(t',t'')+d(t,t'')+d(t',t)}{2} \quad (\forall t \in T, \forall s \in S \setminus \{\hat{s}\}).$$

Then, the total traveling distance satisfies Equation (3) and thus the objective function value of the MIN RES CUT with respect to $\ell(e)$ is equivalent to the total traveling distance.

Finally, we briefly describe an SDP relaxation problem and a randomized algorithm for MIN RES CUT. For MAX RES CUT, Goemans and Williamson [10] proposed a 0.878-randomized approximation algorithm using semidefinite programming. Here we apply Goemans and Williamson's algorithm to the proposed

MIN RES CUT formulation of the HA-assignment problem. In the following, we explain the procedure. The algorithm consists of the following three steps.

1. Semidefinite Programming
   For a given instance of MIN RES CUT $(V, E, r, \ell, E_{\rm cut})$, let $\boldsymbol{W}$ be a matrix whose rows and columns are indexed by V such that $W_{ij} = W_{ji} = w(\{i,j\})$ if $\{i,j\} \in E$, otherwise $W_{ij} = W_{ji} = 0$. Then solve the following semidefinite programming problem:

$$\begin{aligned} \text{minimize} \quad & \sum_i \sum_j C_{ij} X_{ij} \\ \text{subject to} \quad & \boldsymbol{X}_{ii} = 1 \qquad (\forall i \in V), \\ & \boldsymbol{X}_{ij} = -1 \quad (\forall \{i,j\} \in E_{\rm cut}), \\ & \boldsymbol{X} \succeq \boldsymbol{O},\ \boldsymbol{X} \text{ is symmetric},\ \boldsymbol{X} \in \mathrm{R}^{V \times V}, \end{aligned}$$

   where $\boldsymbol{C} = (\mathrm{diag}(\boldsymbol{W}\boldsymbol{e}) - \boldsymbol{W})/4$.
2. Cholesky Decomposition
   Decompose an (almost) optimal solution $\boldsymbol{X}_0$ of the semidefinite programming problem in Step 1 into a matrix $\widehat{\boldsymbol{X}}$ such that $\boldsymbol{X}_0 = \widehat{\boldsymbol{X}}^\top \widehat{\boldsymbol{X}}$ (Cholesky decomposition).
3. Hyperplane Separation
   Generate a vector $\boldsymbol{u}$ at uniformly random on the surface of d-dimensional unit ball and put $V_1 = \{i \in V : \boldsymbol{u}^\top \widehat{\boldsymbol{x}}_i \geq 0\}$ where d is the number of rows of $\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{x}}_i$ is the column vector of $\widehat{\boldsymbol{X}}$ index by $i \in V$. Output a vertex subset $V' = \begin{cases} V_1 & (\text{if } r \notin V_1), \\ V \setminus V_1 & (\text{if } r \in V_1). \end{cases}$

The above three steps terminate in polynomial time. Note that a practical procedure to obtain a good solution is to repeat Step 3 a number of times and output a solution with the best objective value.

Goemans and Williamson [10] showed that the maximization version of the above algorithm finds a feasible solution of MAX RES CUT, and its expected objective value is at least 0.87856 times the optimal value. In case of MIN RES CUT, any non-trivial bound of approximation ratio of the above algorithm is not known.

6 Computational Experiments

In this section, we report our computational results. Tables 1 and 2 shows the approximation ratios, Tables 3 and 4 are the results of CPU time in seconds of the weighted case and the constant case, respectively. Computational experiments were performed as follows.

Each of Tables 1, 2, 3 and 4 shows the results when we generated ten timetables for each size of $2n = 16, 18, 20, 22, 24, 26, 30, 40$. We constructed timetables of a round-robin tournament by the method described in [7]. We used

the distance matrix of TSP instance `att48` from TSPLIB [21]. We chose cities of `att48` with indices from 1 to $2n$. For each instance, we applied the algorithms described in Section 4.1 and generated HA-assignments *upp_itr* times, where $upp\_itr = \min\{\max\{2^{n+1}, 1000\}, 10000\}$. We also applied Goemans and Williamson's SDP based algorithm described in Section 5 and generated 10000 HA-assignments by executing the hyperplane separation procedure 10000 times. Finally, for each algorithm we output a solution with the best objective value. In order to evaluate the quality of the best solutions, we solved the same instances with integer programming in a similar formulation as Trick [17].

Computations were performed on the following softwares and machines; for semidefinite programming problems, we used SDPA 6.0 [20] on Dell Dimension 8100 (CPU: Pentium 4, 1.4 GHz, RAM: 768 MB, OS: Vine Linux 2.6), and for linear programming problems and integer programming problems, we used XPRESS-MP Workstation (Model Builder 10.04, Integer Optimiser 10.27) [3] and CPLEX 8.0 [11], respectively, on Dell Dimension 8250 (CPU: Pentium 4, 3.06 GHz, RAM: 512 MB, OS: Vine Linux 2.6). We did not solve integer programs for $2n = 20$ to 40 in the constant case because it would not terminate within reasonable computational time. In Tables 1 and 2, we summarize the average of ratios of 'the LP optimal value' and 'the objective function value of the best solutions' for each algorithm, where the ratios are described with parentheses.

Weighted Case: Table 1 shows that all of the average of approximation ratios of our three algorithms are less than 1.01. When $2n = 16, 26$, LP relaxation problems give 0-1 valued solution. The notable points are:
(1) our first algorithm can generate solutions whose ratios are better than those of others including the SDP based approach for any number of teams;
(2) randomized rounding algorithms (A1, A2, A3) based on LP relaxation give more acceptable ratios even by the little difference compared with the SDP based approach.
In our computational experiments, we executed each rounding procedure several times and output the best of generated solutions. Thus, outputs depend not only on the expectation but also on the distribution of the objective function value of a generated solution. Depending on the structure of the set of variables with value 1/2, there is possibility that our first algorithm A1 with independent rounding procedure performs better than the other two algorithms; this is because the set of solutions (HA-assignments) that could be generated by A1 includes the set of solutions that could be generated by A2 or A3.
Constant Case: Table 2 shows that almost all of the average of approximation ratios of our randomized rounding algorithms (A1, A2, A3) are less than 1.20, when $2n = 16, 18$. Contrary to the weighted case, the effectiveness of our third algorithm is now emphasized. However, the SDP based approach gives solutions of higher quality.
Half Integrality: As we showed in Theorem 4, **LP** has an optimal solution satisfying half-integrality on $\boldsymbol{y}$. In Tables 1 and 2, **half int.** shows the ratios of the number of variables whose values are 1/2. In the weighted case, almost all

variables are either 0 or 1. In the constant case, all variables take 1/2 as shown in Theorem 5.

CPU time: For the CPU time in Tables 3 and 4, LP based algorithms are much faster than the SDP based approach and integer programs. For instance, in the weighted case of $2n = 16$, the SDP based approach and integer programs took more than 21 seconds and 65 seconds in average, respectively, while LP based algorithms spent less than 1 second. Moreover, LP based algorithms terminated less than 8 seconds for any number of teams in the weighted case. Since Theorem 5 gives a unique optimal solution to **LP** explicitly, we need not to solve **LP** numerically in the constant case. We solved **LP** numerically for comparing with the weighted case. Table 4 shows that LP based algorithm terminated less than 13 seconds in the constant case.

From the overall, we conclude that in the weighted case, LP based algorithms are highly efficient in terms of both quality of solutions and computational speed and SDP based algorithm finds better solutions in the constant case.

Appendix

We prove Theorem 4.

Let **LP** be a linear relaxation of the problem **IP**:

(LP)

$$\text{min.} \quad \sum_{t\in T}\left\{\sum_{s\in\{1,\hat{s}\}} d(t,\tau(t,s))\,y_{t,s} + \sum_{s\in S\setminus\{\hat{s}\}} w_{t,s}\right\}$$

$$\text{s. t.} \quad w_{t,s} \geq d(t',t)\,y_{t,s} + (d(t',t'') - d(t',t))\,y_{t,s+1} \quad \begin{pmatrix}\forall (t,s)\in T\times S\setminus\{\hat{s}\},\ \text{where}\\ t'=\tau(t,s)\ \text{and}\ t''=\tau(t,s+1))\end{pmatrix},$$

$$w_{t,s} \geq (d(t',t'') - d(t,t''))\,y_{t,s} + d(t,t'')\,y_{t,s+1} \quad \begin{pmatrix}\forall (t,s)\in T\times S\setminus\{\hat{s}\},\ \text{where}\\ t'=\tau(t,s)\ \text{and}\ t''=\tau(t,s+1))\end{pmatrix},$$

$$y_{t,s} + y_{\tau(t,s),s} = 1 \qquad (\forall (t,s)\in T\times S),$$

$$y_{t,s} \geq 0 \qquad (\forall (t,s)\in T\times S).$$

It is enough to show that any optimal solution in which $\boldsymbol{y}$ is not half-integral can be expressed as a convex combination of mutually distinct feasible solutions of **LP**. Assume that $(\boldsymbol{y}^*, \boldsymbol{w}^*)$ is an optimal solution in which $\boldsymbol{y}^*$ is not half-integral. By the assumption, there exists at least one element of $\boldsymbol{y}^*$ that is less than 1/2 and exists at least one element more than 1/2. We introduce two functions $g^1_{t,s}(y,y')$ and $g^2_{t,s}(y,y')$ defined as follows:

$$g^1_{t,s}(y,y') = d(t',t)\,y + (d(t',t'') - d(t',t))\,y',$$
$$g^2_{t,s}(y,y') = (d(t',t'') - d(t,t''))\,y + d(t,t'')\,y',$$

where $t' = \tau(t,s)$ and $t'' = \tau(t,s+1)$. It should be noted that $g^1_{t,s}(y,y')$ and $g^2_{t,s}(y,y')$ correspond to the right hand sides of the first and second constraints of **LP**, respectively.

For a sufficiently small positive number ε we construct two vectors $(\boldsymbol{y}^+, \boldsymbol{w}^+)$ and $(\boldsymbol{y}^-, \boldsymbol{w}^-)$ as follows: for each $(t,s) \in T \times S$, we set

$$y^+_{t,s} = \begin{cases} y^*_{t,s} + \varepsilon & (\text{if } \ 0 < y^*_{t,s} < 1/2), \\ y^*_{t,s} - \varepsilon & (\text{if } \ 1/2 < y^*_{t,s} < 1), \\ y^*_{t,s} & (\text{if } \ y^*_{t,s} \in \{0, 1/2, 1\}), \end{cases}$$

$$y^-_{t,s} = \begin{cases} y^*_{t,s} - \varepsilon & (\text{if } \ 0 < y^*_{t,s} < 1/2), \\ y^*_{t,s} + \varepsilon & (\text{if } \ 1/2 < y^*_{t,s} < 1), \\ y^*_{t,s} & (\text{if } \ y^*_{t,s} \in \{0, 1/2, 1\}); \end{cases}$$

and for each $(t,s) \in T \times S \setminus \{\hat{s}\}$, we set

$$w^+_{t,s} = \max\{g^1_{t,s}(y^+_{t,s}, y^+_{t,s+1}),\ g^2_{t,s}(y^+_{t,s}, y^+_{t,s+1})\},$$
$$w^-_{t,s} = \max\{g^1_{t,s}(y^-_{t,s}, y^-_{t,s+1}),\ g^2_{t,s}(y^-_{t,s}, y^-_{t,s+1})\}.$$

Clearly, $y^+_{t,s} + y^+_{\tau(t,s),s} = y^-_{t,s} + y^-_{\tau(t,s),s} = 1$ $(\forall (t,s) \in T \times S)$ and $\boldsymbol{y}^+ \neq \boldsymbol{y}^-$ hold. Choosing ε small enough, we can ensure $\mathbf{0} \leq \boldsymbol{y}^+$ and $\mathbf{0} \leq \boldsymbol{y}^-$. Hence, $(\boldsymbol{y}^+, \boldsymbol{w}^+)$ and $(\boldsymbol{y}^-, \boldsymbol{w}^-)$ are a pair of mutually distinct feasible solutions of **LP**.

From the definition, $(\boldsymbol{y}^+ + \boldsymbol{y}^-)/2 = \boldsymbol{y}^*$. In the following, to prove $(\boldsymbol{w}^+ + \boldsymbol{w}^-)/2 = \boldsymbol{w}^*$ we show that $(w^+_{t,s} + w^-_{t,s})/2 = w^*_{t,s}$ holds for any $(t,s) \in T \times S \setminus \{\hat{s}\}$. Note that the distance matrix satisfies triangle inequalities, i.e., $d_{\tau(t,s),t} + d_{t,\tau(t,s+1)} \geq d_{\tau(t,s),\tau(t,s+1)}$. Since we have

$$g^1_{t,s}(y,y') - g^2_{t,s}(y,y') = (d_{\tau(t,s),t} + d_{t,\tau(t,s+1)} - d_{\tau(t,s),\tau(t,s+1)})(y - y'),$$

the following relationship holds:

$$y \leq y' \implies g^1_{t,s}(y,y') \leq g^2_{t,s}(y,y'),$$
$$y \geq y' \implies g^1_{t,s}(y,y') \geq g^2_{t,s}(y,y').$$

Now we show that $(w^+_{t,s} + w^-_{t,s})/2 = w^*_{t,s}$ holds for any $(t,s) \in T \times S \setminus \{\hat{s}\}$ in each of the following three cases. Note that $w^*_{t,s} = \max\{g^1_{t,s}(y^*_{t,s}, y^*_{t,s+1}),\ g^2_{t,s}(y^*_{t,s}, y^*_{t,s+1})\}$ because $(\boldsymbol{y}^*, \boldsymbol{w}^*)$ is an optimal solution.

Case 1: $y^*_{t,s} < y^*_{t,s+1}$ Choosing ε small enough we can ensure $y^+_{t,s} < y^+_{t,s+1}$ and $y^-_{t,s} < y^-_{t,s+1}$. Then, all of $w^*_{t,s}, w^+_{t,s}$ and $w^-_{t,s}$ are defined by $g^2_{t,s}$. Since $(\boldsymbol{y}^+ + \boldsymbol{y}^-)/2 = \boldsymbol{y}^*$ and $g^2_{t,s}(y,y')$ is a linear function of y and y', the following equalities hold:

$$\begin{aligned} &(1/2)(w^+_{t,s} + w^-_{t,s}) \\ &= (1/2)(g^2_{t,s}(y^+_{t,s}, y^+_{t,s+1}) + g^2_{t,s}(y^-_{t,s}, y^-_{t,s+1})) \\ &= (1/2) g^2_{t,s}(y^+_{t,s} + y^-_{t,s}, y^+_{t,s+1} + y^-_{t,s+1}) \\ &= (1/2) g^2_{t,s}(2y^*_{t,s}, 2y^*_{t,s+1}) = g^2_{t,s}(y^*_{t,s}, y^*_{t,s+1}) \\ &= w^*_{t,s}. \end{aligned}$$

Case 2: $y^*_{t,s} > y^*_{t,s+1}$ Choosing ε small enough we can ensure $y^+_{t,s} > y^+_{t,s+1}$ and $y^-_{t,s} > y^-_{t,s+1}$. Then, all of $w^*_{t,s}, w^+_{t,s}$ and $w^-_{t,s}$ are defined by $g^1_{t,s}$. Since $(\boldsymbol{y}^+ + \boldsymbol{y}^-)/2 = \boldsymbol{y}^*$ and $g^1_{t,s}(y, y')$ is a linear function of y and y', the equality $(w^+_{t,s} + w^-_{t,s})/2 = w^*_{t,s}$ holds.

Case 3: $y^*_{t,s} = y^*_{t,s+1}$ In this case, we have $y^+_{t,s} = y^+_{t,s+1}$ and $y^-_{t,s} = y^-_{t,s+1}$. Thus, the equalities

$$
\begin{aligned}
g^1_{t,s}(y^*_{t,s}, y^*_{t,s+1}) &= g^2_{t,s}(y^*_{t,s}, y^*_{t,s+1}),\\
g^1_{t,s}(y^+_{t,s}, y^+_{t,s+1}) &= g^2_{t,s}(y^+_{t,s}, y^+_{t,s+1}),\\
g^1_{t,s}(y^-_{t,s}, y^-_{t,s+1}) &= g^2_{t,s}(y^-_{t,s}, y^-_{t,s+1})
\end{aligned}
$$

hold. Hence we can consider that all of $w^*_{t,s}, w^+_{t,s}$ and $w^-_{t,s}$ are defined by $g^1_{t,s}$ (and/or $g^2_{t,s}$). Since $(\boldsymbol{y}^+ + \boldsymbol{y}^-)/2 = \boldsymbol{y}^*$ and $g^1_{t,s}(y, y')$ is a linear function of y and y', the equality $(w^+_{t,s} + w^-_{t,s})/2 = w^*_{t,s}$ holds.

From the three cases above, we have $(w^+_{t,s} + w^-_{t,s})/2 = w^*_{t,s}$ for any $(t, s) \in T \times S \setminus \{\hat{s}\}$. Consequently, $(\boldsymbol{w}^+ + \boldsymbol{w}^-)/2 = \boldsymbol{w}^*$ holds. Thus, $(\boldsymbol{y}^*, \boldsymbol{w}^*)$ can be expressed as a convex combination of a mutually distinct pair $(\boldsymbol{y}^+, \boldsymbol{w}^+)$ and $(\boldsymbol{y}^-, \boldsymbol{w}^-)$ of feasible solutions of **LP**. We therefore obtain Theorem 4.

References

1. Avidor, A., Zwick, U.: Approximating MIN k-SAT, In: Algorithms and Computation (ISAAC 2002), Lecture Notes in Computer Science, **2518** (2002) 465–475.
2. Bertsimas, D., Teo, C., Vohra, R.: On dependent randomized rounding algorithms, Operations Research Letters, **24** (1999) 105–114.
3. Dash Associates. XPRESS-MP User Guide and Reference Manual. Dash Associates, 1997.
4. de Werra, D.: Geography, games and graphs, Discrete Applied Mathematics **2** (1980) 327–337.
5. Easton, K., Nemhauser, G.L., Trick, M.A.: Solving the travelling tournament problem: a combined integer programming and constraint programming approach, In: Practice and Theory of Automated Timetabling IV (PATAT 2002), Lecture Notes in Computer Science, **2740** (2003) 100–109.
6. Easton, K., Nemhauser, G.L., Trick, M.A.: Sports scheduling, In: Handbook of Scheduling: Algorithms, Models, and Performance Analysis, Leung, J.Y-T., Anderson, J.H., (eds.), Chapman & Hall, 2004.
7. Elf, M., Jünger, M., Rinaldi, G.: Minimizing breaks by maximizing cuts, Operations Research Letters, **31** (2003) 343–349.
8. Garey, M.R., Johnson, D.S.: Computers and Intractability; a Guide to the Theory of NP-completeness, W. H. Freeman, New York, 1979.
9. Goemans, M.X., Williamson, D.P.: New $\frac{3}{4}$-approximation algorithms for the maximum satisfiability problem, SIAM Journal on Discrete Mathematics, **7** (1994) 656–666.
10. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, Journal of the ACM, **42** (1995) 1115–1145.

11. ILOG: ILOG CPLEX 8.0 User's Manual. ILOG, 2002.
12. Miyashiro, R., Matsui, T.: A polynomial-time algorithm to find an equitable home–away assignment, Operations Research Letters, **33** (2005) 235–241.
13. Miyashiro, R., Matsui, T.: Semidefinite programming based approaches to the break minimization problem, Computers and Operations Research, **33** (2006) 1975–1982.
14. Nemhauser, G.L., Trick, M.A.: Scheduling a major college basketball conference, Operations Research, **46** (1998) 1–8.
15. Régin, J.-C.: Minimization of the number of breaks in sports scheduling problems using constraint programming, In: Constraint Programming and Large Scale Discrete Optimization, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, **57** (2001) 115–130.
16. Russell, R.A., Leung, J.M.Y.: Devising a cost effective schedule for a baseball league, Operations Research, **42** (1994) 614–625.
17. Trick, M.A.: A schedule-then-break approach to sports timetabling, In: Practice and Theory of Automated Timetabling III (PATAT 2000), Lecture Notes in Computer Science, **2079** (2001) 242–253.
18. Suzuka, A., Miyashiro, R., Yoshise, A., Matsui, T.: Semidefinite programming based approaches to home-away assignment problems in sports scheduling, In: The First International Conference on Algorithmic Applications in Management (AAIM 2005), Lecture Notes in Computer Science, **3521** (2005) 95–103.
19. Suzuka, A., Miyashiro, R., Yoshise, A., Matsui, T.: Dependent randomized rounding to the home-away assignment problem in sports scheduling, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (to appear).
20. Yamashita, M., Fujisawa, K., Kojima, M.: Implementation and evaluation of SDPA 6.0, Optimization Methods and Software **18** (2003) 491–505.
21. TSPLIB web page, available at the URL
`http://www.crpc.rice.edu/softlib/tsplib/`
22. Urrutia, S., Ribeiro, C.C.: Maximizing breaks and bounding solutions to the mirrored traveling tournament problem, Discrete Applied Mathematics (to appear).

Table 1. Approximation ratios of the weighted case

	LP		A1	A2	A3	CUT
$2n$	ratio	half int.	ratio	ratio	ratio	ratio
16	1.00000	0.00000	1.00000	1.00000	1.00000	1.00158
18	0.99998	0.01307	1.00075	1.00246	1.00121	1.00295
20	0.99992	0.02158	1.00092	1.00282	1.00184	1.00236
22	1.00000	0.01688	1.00001	1.00329	1.00072	1.00385
24	1.00000	0.00471	1.00001	1.00000	1.00015	1.00423
26	1.00000	0.00000	1.00000	1.00000	1.00000	1.00357
30	0.99969	0.03172	1.00359	1.00875	1.00496	1.00635
40	0.99994	0.00654	1.00017	1.00187	1.00047	1.01007

Table 2. Approximation ratios of the constant case

	LP		A1	A2	A3	CUT
$2n$	ratio	half int.	ratio	ratio	ratio	ratio
16	0.88831	1.00000	1.19226	1.15681	1.07847	1.00138
18	0.88831	1.00000	1.21044	1.15005	1.06241	1.00205
20	(1)	1.00000	(1.36700)	(1.28850)	(1.22000)	(1.13200)
22	(1)	1.00000	(1.37355)	(1.30248)	(1.21240)	(1.13388)
24	(1)	1.00000	(1.38330)	(1.29931)	(1.21667)	(1.13924)
26	(1)	1.00000	(1.38817)	(1.31124)	(1.21746)	(1.14941)
30	(1)	1.00000	(1.40467)	(1.30378)	(1.22533)	(1.15067)
40	(1)	1.00000	(1.42725)	(1.30700)	(1.22800)	(1.15688)

Table 3. CPU time [s] for the weighted case

	A1		A2		A3		CUT		IP	
$2n$	ave.	s. d.	ave.	s. d.	ave.	s. d.	ave.	s. d.	ave.	s. d.
16	0.042	0.0042	0.103	0.0082	0.115	0.0085	24.829	0.6830	0.779	0.1370
18	0.055	0.0097	0.136	0.0165	0.162	0.0079	39.254	0.6962	1.379	0.0348
20	0.112	0.0063	0.315	0.0127	0.409	0.0110	65.079	1.7357	2.194	0.0448
22	0.274	0.0070	0.760	0.0156	0.945	0.0172	99.201	1.9557	3.433	0.0150
24	0.628	0.0063	1.747	0.0241	2.109	0.0050	145.823	3.7068	5.599	0.2223
26	0.897	0.0106	2.517	0.0231	3.192	0.0469	224.273	10.2988	7.308	0.2204
30	1.205	0.0097	3.453	0.1302	3.854	0.2070	411.561	7.7994	13.855	0.3937
40	2.193	0.0206	6.240	0.0501	7.766	0.1773	1955.173	26.2481	52.991	0.3504

Table 4. CPU time [s] for the constant case

	A1		A2		A3		CUT		IP	
$2n$	ave.	s. d.	ave.	s. d.	ave.	s. d.	ave.	s. d.	ave.	s. d.
16	0.042	0.0042	0.162	0.0042	0.179	0.0088	21.701	0.4570	65.900	66.1060
18	0.053	0.0048	0.212	0.0042	0.245	0.0053	32.844	0.7563	2737.900	4999.0000
20	0.119	0.0032	0.520	0.0047	0.613	0.0067	53.550	1.1190	–	–
22	0.278	0.0042	1.267	0.0048	1.438	0.0169	82.185	1.4721	–	–
24	0.648	0.0042	2.997	0.0048	3.347	0.0330	120.208	3.1711	–	–
26	0.926	0.0053	4.330	0.0047	5.004	0.0375	189.170	6.6839	–	–
30	1.242	0.0042	5.840	0.0082	5.964	0.0201	399.349	7.1816	–	–
40	2.227	0.0082	10.739	0.0120	12.471	0.0574	2157.351	69.2466	–	–

$2n$: the number of teams;
ratio: average of ratios of 'the optimal value of **IP**' and 'the objective function value of the best solutions'; digits in parenthesis denote the average of ratios with 'the optimal value of **LP**' instead of 'the optimal value of **IP**';
half int.: ratio of the number of variables whose value is 1/2;
A1, A2, A3: Algorithms A1, A2 and A3 described in Section 4.1;
CUT: SDP based approach described in Section 5;
IP : the integer program in a similar formulation as Trick [17];
avg.: average; **s. d.**: standard deviation.