Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 3, No. 1, pp. 113 – 133, January 2007

THE HOME-AWAY ASSIGNMENT PROBLEMS AND BREAK MINIMIZATION/MAXIMIZATION PROBLEMS IN SPORTS SCHEDULING

Ayami Suzuka, Ryuhei Miyashiro*, Akiko Yoshise† and Tomomi Matsui‡

Dedicated to Professor Masakazu Kojima on the occasion of his 60th birthday.

Abstract: Suppose that we have a timetable of a round-robin tournament with a number of teams, and distances between their homes. The home-away assignment problem is to find a home-away assignment that minimizes the total traveling distance of the teams; the break minimization/maximization problem is to find 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. The aim of this paper is to give a unified view to the three problems. We see that optimal solutions of the break minimization/maximization problems are obtained by solving the home-away assignment problem. For these problems, we propose formulations and approximation preserving reductions, and report known approximation algorithms. For the home-away assignment problem, we give 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. Our computational experiments show that the proposed approaches quickly generate solutions of good approximation ratios.

Key words: *sports scheduling, dependent randomized rounding, semidefinite programming, approximation algorithm, MAX CUT*

Mathematics Subject Classification: *90-08, 90C11, 90C22, 90C27, 90C59*

---

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 (see [15] for comprehensive survey). The home-away

*The author was partially supported by the Grant-in-Aid for Young Scientists (B) 18710127 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

†The author was partially supported by the Grant-in-Aid for Scientific Research (C) 18560052 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

‡The author was partially supported by the Grants-in-Aid for Scientific Research on Priority Areas 16092101 and for Exploratory Research 18651076 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

assignment problem is to assign home or away to each match of a given timetable of round-robin tournament to minimize the total traveling distance of teams in the tournament; the break minimization/maximization problem is to find a home-away assignment that minimizes/maximizes the number of breaks, i.e., consecutive pairs of home games or away games.

Part of the aim of this paper is to give a unified view to the three problems. In [11], Miyashiro and Matsui showed that the break minimization problem and the break maximization problem are essentially equivalent in the following sense; if an optimal solution of an instance of the break minimization/maximization problem is obtained, an optimal solution of the same instance of the break maximization/minimization problem can be directly constructed. In a recently published paper [22], Urrutia and Ribeiro showed the equivalence between the break maximization problem and the constant case home-away assignment problem, which is defined precisely in Section 2. For the break minimization problem, de Werra gave a tight lower bound in his classical paper [4] published in 1980. Therefore, the equivalence proved by Urrutia and Ribeiro implies that a similar bound can be obtained for the constant case home-away assignment problem. Based on the insight, a tight lower bound of the constant case home-away assignment problem was shown by Urrutia and Ribeiro [22]. Elf, Jünger and Rinaldi [7] conjectured that a problem deciding whether a given timetable has a consistent home-away assignment that attains de Werra's lower bound is polynomially solvable. Their conjecture was affirmatively proved by Miyashiro and Matsui in [11]. For the break maximization problem, Miyashiro and Matsui [11] showed that every instance has an approximately good solution. We sum up these results, independently obtained for each problem, to four theorems (Theorems 2.3, 2.4, 2.5 and 2.6) in Section 2.

The three problems discussed in Section 2 are equivalent when we discuss an optimal solution. However, if we discuss approximate solutions, these problems have different features. When we intend to use an existing approximation algorithm without loosing a theoretical approximation ratio, we need to transform a given problem preserving objective values to an instance that is tractable by the algorithm. Note that the break minimization problem is formulated as MAX CUT in [7], and as MAX 2SAT and MAX RES CUT in [12]. Unfortunately, these transformations change objective values, and theoretical approximation ratios can not be expected.

In Section 3 of this paper, we describe approximation preserving reductions of the constant case home-away assignment problem to MIN 2SAT and the break minimization/ maximization problem to UNWEIGHTED MIN/MAX RES CUT. We also report known approximation algorithms for these problems.

In Section 4, we deal with the home-away assignment problem, which includes the break minimization/maximization problems as special cases. A unified view obtained in Section 2 indicates that we can construct a good algorithm by modifying approximation algorithms reported in Section 3. In Section 4.1, we modify the (3/2)-approximation algorithm for MIN 2SAT proposed by Bertsimas, Teo and Vohra [2]. In Section 4.2, we construct a specified graph and employ Goemans and Williamson's algorithm for MAX RES CUT [9].

In Section 5, we give comprehensive comparison of computational efficiency of our algorithms proposed in Section 4.

2 Home-Away Assignment Problem and Break Minimization/ Maximization Problems

Throughout this paper, we deal with a round-robin tournament with the following properties:

$T\backslash S$	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	$\underline{\text{H}}$	$\underline{\text{H}}$	A	H	A	$\underline{\text{A}}$
2	A	H	A	H	A	H	A
3	A	$\underline{\text{A}}$	$\underline{\text{A}}$	H	$\underline{\text{H}}$	$\underline{\text{H}}$	A
4	H	A	H	$\underline{\text{H}}$	A	$\underline{\text{A}}$	H
5	A	$\underline{\text{A}}$	H	A	$\underline{\text{A}}$	$\underline{\text{A}}$	H
6	H	$\underline{\text{H}}$	A	$\underline{\text{A}}$	H	$\underline{\text{H}}$	A
7	H	A	$\underline{\text{A}}$	H	A	$\underline{\text{A}}$	H
8	A	H	$\underline{\text{H}}$	A	H	$\underline{\text{H}}$	$\underline{\text{H}}$

Figure 1: A timetable and HA-assignment of eight teams

- 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.

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

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 T of teams and the set S of slots, 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 an HA-assignment $\mathcal{A}$ 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.

In the following, we give mathematical definitions of the home-away assignment problem and the break minimization/maximization problem. First, we introduce the home-away assignment problem.

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 the home of team t, 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 authors discussed the home-away assignment problem in [18, 19]. A set of instances of the home-away assignment problem satisfying that all the non-diagonal elements of $\mathcal{D}$ are 1 is called the *constant case*. The constant case home-away assignment problem was discussed by Urrutia and Ribeiro [22] in detail.

Next, we give a definition of the 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 [13], 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.

The break minimization problem is well known in sports scheduling. See papers [7, 11, 12, 14, 17] for example. The break maximization and its variants are discussed in [5, 11, 16, 22].

In the rest of this section, we discuss relations among the break minimization problem, the break maximization problem, and the constant case home-away assignment problem. Let the number of breaks in a home-away assignment $\mathcal{A}$ be $b(\mathcal{A})$. In the constant case home-away assignment problem, denote the total traveling distance with respect to $\mathcal{A}$ by $w(\mathcal{A})$.

In [11], Miyashiro and Matsui proved the following lemma, which shows that the break minimization and maximization problems are 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 $s = 1, 3, \dots, 2n-1$, $\widetilde{a}_{t,s} := a_{t,s}$ $(\forall t \in T)$;
- for $s = 2, 4, \dots, 2n-2$, 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 if $\mathcal{A}$ 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 2.1 [11] *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, where $b(\mathcal{A})$ denotes the number of breaks in a home-away assignment $\mathcal{A}$.*

Recently, Urrutia and Ribeiro [22] discussed the constant case home-away assignment problem, and showed the following equality.

Lemma 2.2 [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.*

The above lemma implies the equivalence between the constant case home-away assignment problem and the break maximization problem.

Combining Lemmas 2.1 and 2.2, we have the following theorem showing the equivalence among the three problems.

Theorem 2.3 [11, 22] *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 $b(\widetilde{\mathcal{A}})$.*

In 1980, de Werra [4] proved that an HA-assignment of $2n$ teams that is consistent to a timetable has at least $2n-2$ breaks. This lower bound of the break minimization problem and the above lemmas imply the following inequalities.

Theorem 2.4 [4, 11, 22] *Every feasible HA-assignment $\mathcal{A}$ of $2n$ teams satisfies that $2n-2 \leq b(\mathcal{A}) \leq (2n-1)(2n-2)$ and $w(\mathcal{A}) \geq (2n-1)(n+1)$.*

The above lower bound of the constant case home-away assignment problem was obtained by Urrutia and Ribeiro in [22]. It is known that for any $2n > 0$, 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.

The tightness of the inequalities in Theorem 2.4 depends on a given timetable. More precisely, if a given timetable is ill-conditioned, any consistent HA-assignment does not attain the lower and/or upper bounds in Theorem 2.4. Elf, Jünger and Rinaldi [7] conjectured that a problem deciding "whether a given timetable has a consistent HA-assignment $\mathcal{A}$ which attains the lower bound $2n-2$" is polynomially solvable. Their conjecture was proved affirmatively by Miyashiro and Matsui [11]. They proposed an $\mathrm{O}(n^3)$ 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). These results and Lemmas 2.1 and 2.2 lead the following theorem.

Theorem 2.5 [11] *Given a timetable $\mathcal{T}$ of $2n$ teams, the following problems are solvable in* $\mathrm{O}(n^3)$ *time.*

1. *Find a consistent HA-assignment $\mathcal{A}$ satisfying $b(\mathcal{A}) = 2n-2$.*
2. *Find a consistent HA-assignment $\mathcal{A}$ satisfying $b(\mathcal{A}) = (2n-1)(2n-2)$.*
3. *Find a consistent HA-assignment $\mathcal{A}$ satisfying $w(\mathcal{A}) = (2n-1)(n+1)$.*

Even if a given timetable does not have a consistent HA-assignment attaining the upper bound in Theorem 2.4, Miyashiro and Matsui [11] showed that every timetable has an approximately good HA-assignment for the break maximization problem. Combining their results and Lemmas 2.1 and 2.2, we have the following theorem.

Theorem 2.6 [11] *For any 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 $w(\mathcal{A}) \leq (1/2)n(5n-1)$.*

In the rest of this section, we give an alternative proof of Theorem 2.6, which is a slightly different way from that in [11]. To show the theorem, we construct a randomized algorithm for generating an HA-assignment with a particular structure. The proof in [11] is based on a derandomized version of the following algorithm. By adopting randomization technique, our proof becomes much simpler. In the next section, we use the following randomized algorithm as a subprocedure in our algorithm for the home-away assignment problem.

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 $[\forall t \in T,\ \forall s \in \{1,2,\dots,n-1\},\ a'_{t,2s-1} = a'_{t,2s}]$. For each $s \in \{1,2,\dots,n-1\}$, 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 to satisfy $a'_{t,2s-1} = a'_{t,2s}$ for $t \in T$. 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 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 $[\forall t \in T,\ \forall s \in \{1,2,\dots,n-1\},\ a'_{t,2s-1} = a'_{t,2s}]$. Similarly, we can generate an HA-assignment $\mathcal{A}'$ that is consistent with a given timetable and satisfying $[\forall t \in T,\ \forall s \in \{1,2,\dots,n-1\},\ a'_{t,2s} = a'_{t,2s+1}]$.

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:

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

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.

Proof of Theorem 2.6. Assume that an HA-assignment $\mathcal{A}^*$ is obtained by the procedure *Pairing Slots*. For each team t and each slot $s \in \{2,3,4,\ldots,2n-1\}$, team t has a break at slot s with probability $3/4$. Thus, the expected value of the number of breaks is $\mathrm{E}[b(\mathcal{A}^*)] = (3/4)2n(2n-2) = 3n(n-1)$. It implies that 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 2.1 and 2.2. $\square$

The proof of Theorem 2.6 shows that we have a simple approximation algorithm for the break maximization problem and the constant case home-away assignment problem.

Corollary 2.7 *The procedure Pairing Slots is:*

(i) *a* $(3/4)$*-approximation algorithm for the break maximization problem;*

(ii) *a* $(5/4)$*-approximation algorithm for the constant case home-away assignment problem.*

3 Approximation Preserving Reductions

As shown in the previous section, the three problems are equivalent when we discuss an optimal solution. The complexity statuses of the three problems have not yet been determined; Elf, Jünger and Rinaldi [7] conjectured that the break minimization problem is NP-hard. These facts suggest us to reformulate the three problems so that appropriate approximation algorithms can be adopted. In fact, the break minimization problem has been formulated as MAX CUT in [7], and as MAX 2SAT and MAX RES CUT in [12]. However, we should note that the resultant objective values by these formulations are different from that of the original problem, and hence, desired theoretical approximation ratios of existing algorithms will not be expected by their approaches. In this section, we describe approximation preserving reductions of the three problems to MIN 2SAT, MAX RES CUT and MIN RES CUT, and show how to adopt existing approximation algorithms.

3.1 Constant Case HA-Assignment Problem

First, we show that the constant case home-away assignment problem is reducible to 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 $(t,s) \in T \times S$. Using these constraints, we can eliminate exactly half of propositional variables defined above. Thus, in the constant case, the home-way assignment problem becomes an instance of MIN 2SAT without changing its objective value.

Bertsimas, Teo and Vohra [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 home-away assignment problem can be modeled as MIN 2SAT without changing the objective value, 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.

3.2 Break Maximization Problem

We describe a technique to reduce 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 $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ with $\widetilde{V} = \{v_{t,s} : (t,s) \in T \times S\}$ and $\widetilde{E} = \{\{v_{\tau(t,s-1),s-1}, v_{t,s}\} : t \in T,\ s \in S \backslash \{1\}\}$. 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 defined by $\widetilde{G}$ and E_{cut}, i.e., a vertex subset $V' \subseteq \widetilde{V}$ 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 UNWEIGHTED MAX RES CUT instance and the set of consistent HA-assignments. Since for any $(t,s) \in T \times S \setminus \{1\}$, $a'_{t,s-1} = a'_{t,s}$ if and only if $\{v_{\tau(t,s-1),s-1}, v_{t,s}\} \in \delta(V')$, we have $|\delta(V') \cap \widetilde{E}| = b(\mathcal{A}')$. The break maximization problem is therefore formulated as UNWEIGHTED MAX RES CUT without changing objective value.

For (UNWEIGHTED) MAX RES CUT, Goemans and Williamson [9] proposed a 0.878-approximation algorithm, and accordingly the above transformation leads a 0.878-approximation algorithm for the break maximization problem. Elf, Jünger and Rinaldi [7] transformed the break minimization problem to MAX CUT. Since the objective values of their MAX CUT instance is different from that of the original break minimization problem, a direct application of Goemans and Williamson's 0.878-approximation algorithm for MAX CUT does not guarantee the theoretical approximation ratio.

3.3 Break Minimization Problem

It is quite natural to formulate the break minimization problem as an UNWEIGHTED MIN RES CUT instance, by a similar manner to the formulation of the break maximization problem as UNWEIGHTED MAX RES CUT described in the previous subsection. Unfortunately, for the break minimization problem, no approximation algorithm yielding non-trivial approximation ratio is known. Régin [14] solved instances of the break minimization problem 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] solved instances of up to 26 teams by using MAX CUT solver. All of those methods are based on branch-and-bound technique and give an optimal solution. Miyashiro and Matsui [12] proposed an algorithm based on positive semidefinite programming and reported that their algorithm quickly produces solutions of high quality.

4 Algorithms for HA-Assignment Problem

In this section, we deal with the home-away assignment problem, not only instances of the constant case. For the home-away assignment problem, no approximation algorithm (with non-trivial approximation ratio) is known.

In Section 3, we showed that the constant case home-away assignment problem can be formulated as a MIN 2SAT instance with the same objective values. Bertsimas, Teo and Vohra [2] proposed an integer linear programming based (3/2)-approximation algorithm for MIN 2SAT. In Section 4.1, we modify their algorithm and construct integer linear programming based algorithms for the home-away assignment problem.

Goemans and Williamson's algorithm for MAX RES CUT [9] gives a 0.878-approximation algorithm for the break maximization problem. In Section 4.2, we formulate the home-away assignment problem as MIN RES CUT and employ a minimization version of Goemans and Williamson's algorithm for MAX RES CUT.

4.1 Integer Linear Programming Based Algorithms

In this section, we formulate the home-away assignment problem as an integer programming problem and describe algorithms based on randomized rounding technique. Although the following procedure is basically the same as the algorithm previously proposed by the authors [19], Theorem 4.2 for the constant case is new results obtained in this paper, which induces a simple proof of Theorem 4.3.

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 home-away assignment problem as follows:

(IP)

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

$$\begin{aligned}
\text{s. t.} \quad & w_{t,s} \geq d(t',t)\, y_{t,s} + (d(t',t'') - d(t',t))\, y_{t,s+1} \\
& \qquad \begin{pmatrix} \forall (t,s) \in T \times S \setminus \{\hat{s}\}, \text{ where} \\ t' = \tau(t,s) \text{ and } t'' = \tau(t,s+1)) \end{pmatrix}, \\
& w_{t,s} \geq (d(t',t'') - d(t,t''))\, y_{t,s} + d(t,t'')\, y_{t,s+1} \\
& \qquad \begin{pmatrix} \forall (t,s) \in T \times S \setminus \{\hat{s}\}, \text{ where} \\ t' = \tau(t,s) \text{ and } t'' = \tau(t,s+1)) \end{pmatrix}, \\
& y_{t,s} + y_{\tau(t,s),s} = 1 \qquad (\forall (t,s) \in T \times S), \\
& y_{t,s} \in \{0,1\} \qquad (\forall (t,s) \in T \times S),
\end{aligned}$$

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 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 replacing the 0-1 constraints on variables $y_{t,s}$ with non-negativity constraints $y_{t,s} \geq 0$ $(\forall(t,s) \in T \times S)$. We proved the following theorem showing that **LP** has an optimal solution satisfying half-integrality on variables $y_{t,s}$ $(\forall(t,s) \in T \times S)$.

Theorem 4.1 [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* $(t,s) \in T \times S$.

4.1.1 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 (see [19] for detail). 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 [8].

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, Teo and Vohra [2] proposes an approximation algorithm based on randomized rounding introducing dependencies in the rounding process. Our second algorithm is a direct application of their rounding technique, which uses random HA-assignment as an initial solution.

A3: Randomized Rounding with the Procedure Pairing Slots

In our third algorithm, we modify dependent randomized rounding procedure A2 proposed by Bertsimas, Teo and Vohra [2] as follows. We generate an HA-assignment $\mathcal{A}^* = (a^*_{t,s})$ by the procedure Pairing Slots proposed in Section 2, and apply 'Dependent Randomized Rounding' procedure in [2]. To obtain a number of initial HA-assignments $\mathcal{A}^*$, we execute the procedure Pairing Slots several times.

4.1.2 Analysis of Constant Case

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

Theorem 4.2 *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 of Theorem 4.2. It is clear that the solution $(\boldsymbol{y}^*, \boldsymbol{w}^*)$ defined above is a feasible solution for **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

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

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 $(\boldsymbol{y}', \boldsymbol{w}')$. □

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 4.1.1 by $Z^{\mathrm{A1}}, Z^{\mathrm{A2}}$ and Z^{A3}, respectively. Then, the following theorem holds.

Theorem 4.3 [19] *In the constant case, the following holds:*

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

The above theorem indicates that, for the constant case, the algorithm A3 finds a solution whose objective value is better than that of A1 and A2. However, our computational experiments in Section 5 show that, for the *weighted case*, i.e., instances with a general distance matrix, A1 generates solutions with better approximation ratios than A2 or A3 on average.

4.2 Positive Semidefinite Programming Based Algorithm

In this section, we propose a formulation of the home-away assignment problem as MIN RES CUT and a randomized algorithm based on a positive semidefinite programming relaxation. In [12], Miyashiro and Matsui used a similar idea to the break minimization problem, which corresponds to the constant case. In [18], the authors proposed a similar algorithm for the home-away assignment problem of a double round-robin tournament and reported the results of computational experiences. The following procedure corresponds to the case of a single round-robin tournament.

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. The problem MIN RES CUT is defined as follows: given a graph $G = (V, E)$, a specified vertex $r \in V$, a weight function $\ell : E \to \mathbb{R}$, 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 $V'' \subseteq V$, $\delta(V'') = \delta(V \setminus V'')$ holds. However, the redundant condition helps to formulate the home-away 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 [9] proposed a 0.878-approximation algorithm for MAX RES CUT.

Now we formulate the home-away assignment problem as MIN RES CUT. The following procedure is similar to that in Subsection 3.2. Given a timetable $\mathcal{T} = (\tau(t, s))$ $((t, s) \in T \times S)$, we construct an undirected graph $\widehat{G} = (\widehat{V}, \widehat{E})$ by modifying the graph $\widetilde{G}$ defined in Subsection 3.2. We introduce an artificial vertex r and define

$$\begin{aligned} \widehat{V} &= \widetilde{V} \cup \{r\} = \{v_{t,s} : (t, s) \in T \times S\} \cup \{r\}, \\ \widehat{E} &= \{\{v_{t,s-1}, v_{t,s}\} : t \in T,\ s \in S \setminus \{1\}\} \cup \{\{r, v_{t,s}\} : (t, s) \in T \times S\}. \end{aligned}$$

We use the edge set E_{cut} defined in Subsection 3.2. For a feasible solution V' of this MIN RES CUT instance, i.e., a vertex subset $V' \subseteq \widehat{V}$ 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.

In the following, we use the notations $t' = \tau(t,s)$, $t'' = \tau(t,s+1)$. We define a weight function $\ell : \widehat{E} \to \mathbb{R}_+$ as follows:

$$\begin{aligned}
\ell(\{v_{t,s}, r\}) &= \frac{d(t',t'') - d(t,t'') + d(t',t)}{2} \\
&\quad + \frac{d(\tau(t,s-1),t') + d(t,t') - d(\tau(t,s-1),t)}{2} \\
&\quad (\forall t \in T, \forall s \in S \setminus \{1, \hat{s}\}), \\
\ell(\{v_{t,1}, r\}) &= d(t,\tau(t,1)) + \frac{d(\tau(t,1),\tau(t,2)) - d(t,\tau(t,2)) + d(\tau(t,1),t)}{2}, \\
\ell(\{v_{t,\hat{s}}, r\}) &= d(\tau(t,\hat{s}),t) + \frac{d(\tau(t,\hat{s}-1),\tau(t,\hat{s})))}{2} + \frac{d(t,\tau(t,\hat{s})) - d(\tau(t,\hat{s}-1),t)}{2}, \\
\ell(\{v_{t,s}, v_{t,s+1}\}) &= \frac{-d(t',t'') + d(t,t'') + d(t',t)}{2} \qquad (\forall t \in T, \forall s \in S \setminus \{\hat{s}\}).
\end{aligned}$$

Then we can show that the objective function value of the MIN RES CUT with respect to $\ell(e)$ is equivalent to the total traveling distance (see [18, 19] for detail).

Finally, we briefly describe an SDP relaxation problem and a randomized algorithm for MIN RES CUT. For MAX RES CUT, Goemans and Williamson [9] 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 home-away assignment problem. The algorithm consists of the following three steps.

1. Semidefinite Programming

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

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

   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 $V_1 = \{i \in \widehat{V} : \boldsymbol{u}^\top \widehat{\boldsymbol{x}}_i \geq 0\}$ 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 $i \in \widehat{V}$. Output a vertex subset $V' = \begin{cases} V_1 & (\text{if } r \notin V_1), \\ \widehat{V} \setminus V_1 & (\text{if } r \in V_1). \end{cases}$

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.

$\boxed{5}$ Computational Experiments

In this section, we report our computational results. We evaluate the proposed algorithms in terms of approximation ratios and CPU times. Computational experiments were performed as follows.

We generated ten timetables for each size of $2n = 16, 18, 20, 22, 24, 26, 30, 40$. These timetables were created with the method described in [7]. For the weighted case, we used the distance matrix of the TSP instance `att48` from TSPLIB [21], and chose the cities of `att48` with indices from 1 to $2n$. For each instance, we applied the three algorithms described in Section 4.1.1 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 4.2 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 environment: 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 [10], 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.

All the computational results are described in Tables 1–8 in Appendix. In the following, we analyze the obtained results.

Weighted Case:

Table 1 shows that all of the averages of approximation ratios of our three algorithms are less than 1.01. When $2n = 16, 26$, LP relaxation problems give 0-1 valued solutions. The notable points are:
(1) the algorithm A1 can generate solutions whose ratios are better than those of A2, A3 and the SDP based approach for any number of teams;
(2) the randomized rounding algorithms A1, A2 and A3 based on LP relaxation give more acceptable ratios even by the little difference compared with the SDP based approach.

Constant Case:

Table 2 shows that almost all of the averages of approximation ratios of our randomized rounding algorithms A1, A2 and 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 4.1, **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 4.2.

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 about 21 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 4.2 gives a unique optimal solution of **LP** explicitly, we need not to solve **LP** numerically in the constant case. However, we solved **LP** numerically to compare 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 in the constant case, SDP based algorithm finds better solutions.

Mirrored Double Round-Robin Tournament

Lastly, we briefly report the results of computational experiences on a mirrored double round-robin tournament. A double round-robin tournament of $2n$ teams is a set of games with $2(2n-1)$ slots in which every team plays every other team exactly once at home and once at away. A double round-robin tournament is said to be *mirrored* when the first and the second half of the timetable are identical (except the HA-assignment), i.e., only home and away are exchanged. For a precise definition of the home-away assignment problem of a double round-robin tournament, see the paper [18]. The authors proposed a semidefinite programming based algorithm for the home-away assignment problem of a double round-robin tournament [18]. We can also devise a linear programming based algorithm for the mirrored double round-robin case, in a similar way to that described in Section 4.1.1. Here we omit the details.

Tables 5–8 show the results of our computational experiences on a mirrored double round-robin tournament. Different from the single round-robin tournament, even in the weighted case, in optimal solutions of linear relaxation problems almost all variables take the value 1/2. This property comes from the fact that a mirrored double round-robin tournament has a structure with high symmetry. Thus, the SDP based algorithm finds better solutions for both weighted and constant cases.

Appendix

The results of computational experiments are summarized in Tables 1–8. In the tables, each abbreviation means the following:
$2n$: 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 values are 1/2;
A1, A2, A3: the algorithms A1, A2 and A3 described in Section 4.1.1;
CUT: SDP based approach described in Section 4.2;
IP : the integer program in a similar formulation as Trick [17];
avg.: average;
s. d.: standard deviation.

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	0.004	0.103	0.008	0.115	0.009	24.829	0.683	0.779	0.137
18	0.055	0.010	0.136	0.017	0.162	0.008	39.254	0.696	1.379	0.035
20	0.112	0.006	0.315	0.013	0.409	0.011	65.079	1.736	2.194	0.045
22	0.274	0.007	0.760	0.016	0.945	0.017	99.201	1.956	3.433	0.015
24	0.628	0.006	1.747	0.024	2.109	0.005	145.823	3.707	5.599	0.222
26	0.897	0.011	2.517	0.023	3.192	0.047	224.273	10.299	7.308	0.220
30	1.205	0.010	3.453	0.130	3.854	0.207	411.561	7.799	13.855	0.394
40	2.193	0.021	6.240	0.050	7.766	0.177	1955.173	26.248	52.991	0.350

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	0.004	0.162	0.004	0.179	0.009	21.701	0.457	65.900	66.106
18	0.053	0.005	0.212	0.004	0.245	0.005	32.844	0.756	2737.900	4999.000
20	0.119	0.003	0.520	0.005	0.613	0.007	53.550	1.119	–	–
22	0.278	0.004	1.267	0.005	1.438	0.017	82.185	1.472	–	–
24	0.648	0.004	2.997	0.005	3.347	0.033	120.208	3.171	–	–
26	0.926	0.005	4.330	0.005	5.004	0.038	189.170	6.684	–	–
30	1.242	0.004	5.840	0.008	5.964	0.020	399.349	7.182	–	–
40	2.227	0.008	10.739	0.012	12.471	0.057	2157.351	69.247	–	–

Table 5: Approximation ratios of the weighted case (mirrored double round-robin)

	LP		A1	A2	A3	CUT
$2n$	ratio	half int.	ratio	ratio	ratio	ratio
16	0.95004	1.00000	1.27971	1.29520	1.17460	1.00112
18	0.95064	1.00000	1.29357	1.23069	1.16830	1.00141
20	0.95233	0.99895	1.29756	1.22349	1.19202	1.00122
22	0.94706	1.00000	1.29057	1.23147	1.17955	1.00368
24	0.94483	1.00000	1.29720	1.21574	1.16753	1.00478
26	(1)	1.00000	(1.37171)	(1.30406)	(1.23298)	(1.05732)
30	(1)	1.00000	(1.39525)	(1.29799)	(1.23835)	(1.05470)
40	(1)	0.99974	(1.41936)	(1.31181)	(1.23814)	–

Table 6: Approximation ratios of the constant case (mirrored double round-robin)

	LP		A1	A2	A3	CUT
$2n$	ratio	half int.	ratio	ratio	ratio	ratio
16	0.87790	1.00000	1.20226	1.10113	1.07523	1.00398
18	(1)	1.00000	(1.38413)	(1.32063)	(1.22143)	(1.14405)
20	(1)	1.00000	(1.36700)	(1.28850)	(1.22000)	(1.14667)
22	(1)	1.00000	(1.38949)	(1.31410)	(1.23769)	(1.14609)
24	(1)	1.00000	(1.40266)	(1.31897)	(1.23511)	(1.15337)
26	(1)	1.00000	(1.40754)	(1.33137)	(1.23183)	(1.15535)
30	(1)	1.00000	(1.42226)	(1.31876)	(1.24113)	(1.16203)
40	(1)	1.00000	(1.43968)	(1.31791)	(1.23684)	–

Table 7: CPU time [s] for the weighted case (mirrored double round-robin)

	A1		A2		A3		CUT		IP	
$2n$	ave.	s. d.	ave.	s. d.	ave.	s. d.	ave.	s. d.	ave.	s. d.
16	0.063	0.005	0.170	0.000	0.205	0.005	106.307	4.244	13.240	2.087
18	0.083	0.005	0.221	0.003	0.259	0.006	195.777	5.782	24.500	11.341
20	0.186	0.005	0.537	0.005	0.616	0.094	328.678	8.372	328.678	8.372
22	0.432	0.006	1.288	0.004	1.499	0.020	531.031	22.825	1550.260	2235.544
24	1.017	0.007	3.087	0.005	3.620	0.030	810.770	74.605	68341.735	124286.748
26	1.439	0.010	4.441	0.007	5.329	0.033	1192.616	62.123	—	—
30	1.926	0.010	5.908	0.006	7.065	0.059	2721.488	141.783	—	—
40	3.442	0.017	10.725	0.058	12.795	0.049	—	—	—	—

Table 8: CPU time [s] for the constant case (mirrored double round-robin)

	A1		A2		A3		CUT		IP	
$2n$	ave.	s. d.	ave.	s. d.	ave.	s. d.	ave.	s. d.	ave.	s. d.
16	0.067	0.005	0.170	0.000	0.200	0.005	109.039	4.855	8425.125	5888.344
18	0.080	0.000	0.222	0.004	0.260	0.005	187.708	10.673	—	—
20	0.183	0.005	0.548	0.033	0.635	0.034	290.856	3.874	—	—
22	0.432	0.006	1.295	0.005	1.493	0.018	468.013	6.526	—	—
24	1.016	0.007	3.087	0.008	3.626	0.034	748.987	16.614	—	—
26	1.434	0.007	4.441	0.006	5.344	0.028	1140.876	26.951	—	—
30	1.932	0.016	5.923	0.010	7.070	0.072	2526.272	70.011	—	—
40	3.433	0.010	10.735	0.067	12.807	0.022	—	—	—	—

References

[1] A. Avidor and U. Zwick, Approximating MIN k-SAT, in *Algorithms and Computation (ISAAC 2002): Lecture Notes in Comput. Sci. 2518*, P. Bose and P. Morin (eds.), Springer, Berlin, 2002, pp. 465–475.

[2] D. Bertsimas, C. Teo and R. Vohra, On dependent randomized rounding algorithms, *Oper. Res. Lett.* 24 (1999) 105–114.

[3] Dash Associates, *XPRESS-MP User Guide and Reference Manual*, Dash Associates, Blisworth, 1997.

[4] D. de Werra, Geography, games and graphs, *Discrete Appl. Math.* 2 (1980) 327–337.

[5] K. Easton, G.L. Nemhauser and M.A. Trick, 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 Comput. Sci. 2740*, E. Burke and P. De Causmaecker (eds.), Springer, Berlin, 2003, pp. 100–109.

[6] K. Easton, G.L. Nemhauser and M.A. Trick, Sports scheduling, in *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, J.Y-T. Leung (ed.), Chapman & Hall, Boca Raton, 2004, pp. 52-1–52-19.

[7] M. Elf, M. Jünger and G. Rinaldi, Minimizing breaks by maximizing cuts, *Oper. Res. Lett.* 31 (2003) 343–349.

[8] M.X. Goemans and D.P. Williamson, New $\frac{3}{4}$-approximation algorithms for the maximum satisfiability problem, *SIAM J. Discrete Math.* 7 (1994) 656–666.

[9] M.X. Goemans and D.P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM* 42 (1995) 1115–1145.

[10] ILOG, *ILOG CPLEX 8.0 User's Manual*, ILOG, Gentilly, 2002.

[11] R. Miyashiro and T. Matsui, A polynomial-time algorithm to find an equitable home-away assignment, *Oper. Res. Lett.* 33 (2005) 235–241.

[12] R. Miyashiro and T. Matsui, Semidefinite programming based approaches to the break minimization problem, *Comput. Oper. Res.* 33 (2006) 1975–1982.

[13] G.L. Nemhauser and M.A. Trick, Scheduling a major college basketball conference, *Oper. Res.* 46 (1998) 1–8.

[14] J.-C. Régin, Minimization of the number of breaks in sports scheduling problems using constraint programming, in *Constraint Programming and Large Scale Discrete Optimization: DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 57*, E.C. Freuder and R.J. Wallace (eds.), Amer. Math. Soc., Providence, 2001, pp. 115–130.

[15] R.V. Rasmussen and M.A. Trick, Round robin scheduling — a survey, *Working Paper 2006/2*, University of Aarhus, Aarhus, 2006.

[16] R.A. Russell and J.M.Y. Leung, Devising a cost effective schedule for a baseball league, *Oper. Res.* 42 (1994) 614–625.

[17] M.A. Trick, A schedule-then-break approach to sports timetabling, in *Practice and Theory of Automated Timetabling III (PATAT 2000): Lecture Notes in Comput. Sci. 2079*, E. Burke and W. Erben (eds.), Springer, Berlin, 2001, pp. 242–253.

[18] A. Suzuka, R. Miyashiro, A. Yoshise and T. Matsui, 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 Comput. Sci. 3521*, N. Megiddo, Y. Xu and B. Zhu (eds.), Springer, Berlin, 2005, pp. 95–103.

[19] A. Suzuka, R. Miyashiro, A. Yoshise and T. Matsui, Dependent randomized rounding to the home-away assignment problem in sports scheduling, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E89-A (2006) 1407–1416.

[20] M. Yamashita, K. Fujisawa and M. Kojima, Implementation and evaluation of SDPA 6.0, *Optim. Methods Softw.* 18 (2003) 491–505.

[21] TSPLIB web page,
http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

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

Manuscript received 12 January 2006
revised 2 April 2006
accepted for publication 2 November 2006

Ayami Suzuka
Graduate School of Systems and Information Engineering, The University of Tsukuba,
Tsukuba, Ibaraki 305-8573, Japan
E-mail address: `asuzuka@sk.tsukuba.ac.jp`

Ryuhei Miyashiro
Institute of Symbiotic Science and Technology, Tokyo University of Agriculture and Technology,
Koganei, Tokyo 184-8588, Japan
E-mail address: `r-miya@cc.tuat.ac.jp`

Akiko Yoshise
Graduate School of Systems and Information Engineering, The University of Tsukuba,
Tsukuba, Ibaraki 305-8573, Japan
E-mail address: `yoshise@sk.tsukuba.ac.jp`

Tomomi Matsui
Department of Mathematical Informatics, Graduate School of Information Science and Technology,
The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan
Current address:
Department of Information and System Engineering, Faculty of Science and Engineering,
Chuo University, Bunkyo-ku, Tokyo 112-8551, Japan
E-mail address: `matsui@ise.chuo-u.ac.jp`