Department of Social Systems and Management

Discussion Paper Series

No. 1133

Dependent Randomized Rounding to the Home-Away Assignment Problem in Sports Scheduling

Ayami SUZUKA† Ryuhei MIYASHIRO‡
Akiko YOSHISE† Tomomi MATSUI§

September 28, 2005

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

Dependent Randomized Rounding to the Home-Away Assignment Problem in Sports Scheduling *

Ayami SUZUKA† Ryuhei MIYASHIRO‡
Akiko YOSHISE† Tomomi MATSUI§

† Graduate School of Systems and Information Engineering,
University of Tsukuba,
Tsukuba, Ibaraki 305-8573, Japan.
{asuzuka,yoshise}@sk.tsukuba.ac.jp

‡ Institute of Symbiotic Science and Technology,
Tokyo University of Agriculture and Technology,
Koganei, Tokyo 184-8588, Japan.
r-miya@cc.tuat.ac.jp

§ Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan.
http://www.simplex.t.u-tokyo.ac.jp/~tomomi/

September, 2005

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. We propose a formulation of the home-away assignment problem as an integer program, and a rounding algorithm based on Bertsimas, Teo and Vohra's dependent randomized rounding method [2]. Computational experiments show that our method quickly generates feasible solutions close to optimal.
Keywords: sports scheduling; timetabling; approximation algorithm; dependent randomized rounding

*This paper is also referred to as A. Suzuka, R. Miyashiro, A. Yoshise and T. Matsui, Dependent Randomized Rounding to the Home-Away Assignment Problem in Sports Scheduling," METR2005-27, Department of Mathematical Engineering and Information Physics, Faculty of Engineering, The University of Tokyo, 2005.

1 Introduction

Sports scheduling has recently become one of the main topics in the area of scheduling (e.g., see "*Handbook of Scheduling*" Chapter 52 (Sports Scheduling) [5]). This paper deals with the *home-away assignment problem* in sports scheduling, which assigns home or away to each match of a round-robin tournament so as to minimize total traveling distance. We propose an approach based on Bertsimas, Teo and Vohra's dependent randomized rounding method [2] and the home-away assignment generation algorithm proposed in [12].

This paper is organized as follows: Section 2 defines the home-away assignment problem, and proposes formulations of the problem as an integer program; after describing our rounding algorithm in Section 3, we discuss its approximation ratio in Section 4; Section 5 reports the results of computational experiments; Section 6 states conclusions.

In [16], the authors proposed technique to transform the home-away assignment problem to MIN RES CUT, and applied Goemans and Williamson's algorithm for MAX RES CUT [9] based on the positive semidefinite programming relaxation. We have shown that the above approach gives solutions of high quality. However, the computational effort for solving positive semidefinite programming problems is not ignorable. In this paper, we propose algorithms based on linear programming relaxation, and the computational experiments show that our algorithm efficiently produces feasible solutions close to optimal. Surprisingly, in many cases the quality of obtained solutions is competitive with or better than that of solutions obtained by the positive semidefinite programming approach.

For a given timetable, the problem to find a home-away assignment that minimizes the number of breaks (consecutive pairs of home-games or away-games) is called the break minimization problem. There are several previous results on this problem (see [14, 17, 6, 11, 12]). In [12], Miyashiro and Matsui showed that the break minimization problem is essentially equivalent to the problem for minimizing the total traveling distance when the distance of every pair of homes is equal to 1. Thus we can apply the algorithm proposed in this paper to the break minimization problem.

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.

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.

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

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

Figure 1: A timetable and home-away assignment of six teams

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 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. Accordingly, 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 4 in slot 5, and thus team 4 plays team 2 in slot 5.

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 $\forall (t,s) \in T \times S$, $\{a_{t,s}, a_{\tau(t,s),s}\} = \{\text{A}, \text{H}\}$ holds. 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 timetabling, the total traveling distance is required to be reduced [15]. 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.

We propose a formulation of the Home-Away assignment problem as an integer programming problem. 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:

$$
\begin{array}{lll}
\textbf{(IP)} & & \\
\text{min.} & \displaystyle\sum_{t\in T}\left\{\sum_{s\in\{1,\hat{s}\}} d(t,\tau(t,s))\,y_{t,s} + \sum_{s\in S\setminus\{\hat{s}\}} w_{t,s}\right\} & \\
\text{s. t.} & w_{t,s} \geq d(t',t)\,y_{t,s} + (d(t',t'') - d(t',t))\,y_{t,s+1} & \\
 & \quad \left(\begin{array}{l}\forall (t,s) \in T\times S\setminus\{\hat{s}\},\ \text{where} \\ t' = \tau(t,s) \text{ and } t'' = \tau(t,s+1))\end{array}\right), & \\
 & w_{t,s} \geq (d(t',t'') - d(t,t''))\,y_{t,s} + d(t,t'')\,y_{t,s+1} & \\
 & \quad \left(\begin{array}{l}\forall (t,s) \in T\times S\setminus\{\hat{s}\},\ \text{where} \\ t' = \tau(t,s) \text{ and } t'' = \tau(t,s+1))\end{array}\right), & \\
 & y_{t,s} + y_{\tau(t,s),s} = 1 & (\forall (t,s) \in T\times S), \\
 & y_{t,s} \in \{0,1\} & (\forall (t,s) \in T\times S),
\end{array}
$$

where $w_{t,s}$ $((t, s) \in T \times S \setminus \{\hat{s}\})$ are free 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 $y_{t,s} \geq 0$ $(\forall (t, s) \in T \times S)$. The following theorem shows that **LP** has an optimal solution satisfying half-integrality.

Theorem 1 *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 A.

A set of instances satisfying that all the non-diagonal elements of $\mathcal{D}$ are 1 is called the *constant case*. Here we show that in the constant case, **IP** becomes an instance of MIN 2SAT. Given a set of clauses each of which consists of at most two literals, MIN 2SAT finds 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 $y_{t,s} = 1$. Then the traveling distance $w_{t,s}$ is equal to 1 if and only if the clause $Y_{t,s} \vee Y_{t,s+1}$ has the value `TRUE`. We can express the term $d(t, \tau(t,s))\, y_{t,s}$ appearing in the objective function of **IP** as a clause consists of one literal $Y_{t,s}$. The equality $y_{t,s} + y_{\tau(t,s),s} = 1$ is essentially equivalent to the property that $Y_{t,s}$ is the negation of $Y_{\tau(t,s),s}$. See [2, 1] for some results on MIN 2SAT.

3 Randomized Rounding Algorithms

In our algorithms, we solve the linear relaxation problem **LP** first. If an obtained solution is 0-1 valued, we have an optimal solution for 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 **LP** by $(\boldsymbol{y}^*, \boldsymbol{w}^*)$.

3.1 Independent Randomized Rounding

The first algorithm is the *independent randomized rounding algorithm* that generates a 0-1 valued solution as follows. For each pair of teams $\{t, t'\}$, we decide the venue of the match between t and t' independently. Let s be the slot when t and t' plays 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 2SAT proposed by Goemans and Williamson in [8].

3.2 Dependent Randomized Rounding

In Section 2, we described that **IP** becomes an instance of MIN 2SAT in the constant case. For MIN 2SAT, Bertsimas et al. [2] proposes an approximation algorithm based on randomized rounding introducing dependencies in the rounding process. Our second algorithm is a direct application of their algorithm to a general case with a given distance matrix. 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.

<u>Dependent Randomized Rounding</u>

<u>Step 0:</u> Generate a uniform random number $U \in (0,1]$.

<u>Step 1:</u> 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**.

Bertsimas et al. [2] showed that the expected objective value obtained by their rounding method for MIN 2SAT is at most $\frac{3}{2}$ times the optimal value. In the constant case, since the Home-Away assignment problem corresponds to MIN 2SAT, the approximation ratio of the above algorithm is also bounded by $\frac{3}{2}$. 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.

3.3 Generating an Initial HA-assignment

In our third algorithm, we generate an HA-assignment $\mathcal{A}^* = (a^*_{t,s})$ by an algorithm based on the procedure proposed in [12] and execute the 'Dependent Randomized Rounding' procedure described in Section 3.2.

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:

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

Our third algorithm uses an HA-assignment $\mathcal{A}^*$ obtained by the following procedure.

Flipping

Step 0: Execute one of Steps 1 and 2 at random.

Step 1: Generate an HA-assignment $\mathcal{A}' = (a'_{t,s})$ that is consistent with a given timetable and satisfying that $[\forall t \in T,\ \forall s \in \{1, 2, \ldots, n-1\},\ a'_{t,2s-1} = a'_{t,2s}]$. Set $\mathcal{A}^* = (a^*_{t,s})$ be an HA-assignment obtained from $\mathcal{A}'$ by flipping slots $\{2s-1, 2s\}$ with probability $(1/2)$ for each $s \in \{1, 2, \ldots, n-1\}$ independently.

Step 2: Generate an HA-assignment $\mathcal{A}' = (a'_{t,s})$ that is consistent with a given timetable and satisfying $[\forall t \in T,\ \forall s \in \{1, 2, \ldots, n-1\},\ a'_{t,2s} = a'_{t,2s+1}]$. Set $\mathcal{A}^* = (a^*_{t,s})$ be an HA-assignment obtained from $\mathcal{A}'$ by flipping slots $\{2s, 2s+1\}$ with probability $1/2$ for each $s \in \{1, 2, \ldots, n-1\}$ independently.

We need to generate a specified HA-assignment $\mathcal{A}'$ at the beginning of Steps 1 and 2, which is obtained by the method proposed in [12]. We briefly describe the algorithm in Appendix B.

3.4 Derandomization

Here we derandomize the procedure 'Dependent Randomized Rounding,' which extremely shortens practical computational time. Here we assume that the variables $\boldsymbol{y}^*$, which is obtained by solving **LP**, satisfy half-integrality. Then, it is easy to see that if the uniform random number U obtained at Step 0 in the procedure 'Dependent Randomized Rounding' satisfies that $0 < U \leq 1/2$, 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 that $1/2 < U \leq 1$, 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 one of the best solutions. 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 'Dependent Randomized Rounding' procedure. A practical procedure to obtain a better solution in our second algorithm 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 specified HA-assignment $\mathcal{A}'$ by randomly flipping slots $\{2s-1, 2s\}$ in Step 1 and slots $\{2s, 2s+1\}$ in Step 2 for each $s \in \{1, 2, \dots, n-1\}$ several times.

In our computational experiences, we use the above derandomized procedure. We will use the original 'Dependent Randomized Rounding' procedure in the next section to discuss the approximation ratio.

4 Approximation Ratios

In this section, we discuss the approximation ratios of our algorithms in the constant case. We denote the optimal value of **IP** and **LP** by Z^{IP} and Z^{LP}, respectively. We also denote the objective values obtained by our first, second and third algorithms by $Z^{\mathrm{A1}}, Z^{\mathrm{A2}}$ and Z^{A3}, respectively. Then the following theorem holds.

Theorem 2 *In the constant case, the following inequalities hold:*

1. $\mathrm{E}[Z^{\mathrm{A1}}] \leq \frac{3}{2} Z^{\mathrm{LP}} \leq \frac{3}{2} Z^{\mathrm{IP}}$,
2. $\mathrm{E}[Z^{\mathrm{A2}}] \leq \frac{3}{2} Z^{\mathrm{LP}} \leq \frac{3}{2} Z^{\mathrm{IP}}$,
3. $\mathrm{E}[Z^{\mathrm{A3}}] \leq \frac{5}{4} Z^{\mathrm{LP}} \leq \frac{5}{4} Z^{\mathrm{IP}}$.

Proof. In the constant case, it is easy to see that if we fix variables $\boldsymbol{y}$ in **LP**, the objective value obtained by choosing $\boldsymbol{w}$ that minimizes the objective function is

$$Z^{\mathrm{C}}(\boldsymbol{y}) = \sum_{t \in T} \left\{ \sum_{s \in \{1, \hat{s}\}} y_{t,s} + \sum_{s \in S \setminus \{\hat{s}\}} \max\{y_{t,s}, y_{t,s+1}\} \right\}.$$

We denote an optimal solution of **LP** by $(\boldsymbol{y}^*, \boldsymbol{w}^*)$. In this proof, we assume that $\boldsymbol{y}^*$ satisfies half-integrality.

Let $\boldsymbol{y}^{\mathrm{A}}$ be a solution obtained by one of our algorithms. Here we note that $\boldsymbol{y}^{\mathrm{A}}$ is a vector of random variables. Then it is clear that $\mathrm{E}[y^{\mathrm{A}}_{t,s}] = \Pr[y^{\mathrm{A}}_{t,s} = 1] = y^*_{t,s}$. For the second term of the above objective value, we have

$$\begin{aligned} \mathrm{E}[\max\{y^{\mathrm{A}}_{t,s}, y^{\mathrm{A}}_{t,s+1}\}] &\leq \min\{1, \mathrm{E}[y^{\mathrm{A}}_{t,s} + y^{\mathrm{A}}_{t,s+1}]\} \\ &= \min\{1, \mathrm{E}[y^{\mathrm{A}}_{t,s}] + \mathrm{E}[y^{\mathrm{A}}_{t,s+1}]\} \\ &= \min\{1, \Pr[y^{\mathrm{A}}_{t,s} = 1] + \Pr[y^{\mathrm{A}}_{t,s+1} = 1]\} \\ &= \min\{1, y^*_{t,s} + y^*_{t,s+1}\}. \end{aligned}$$

Case a: When at least one of $\{y^*_{t,s}, y^*_{t,s+1}\}$ is 0-1 valued, the property

$$\begin{aligned}\mathrm{E}[\max\{y^{\mathrm{A}}_{t,s}, y^{\mathrm{A}}_{t,s+1}\}] &\leq \min\{1, y^*_{t,s} + y^*_{t,s+1}\} \\ &= \max\{y^*_{t,s}, y^*_{t,s+1}\}\end{aligned}$$

holds.

Case b: We need to consider the case that $(y^*_{t,s}, y^*_{t,s+1}) = (1/2, 1/2)$ for proposed three algorithms, respectively.

Case b-1: Let $\boldsymbol{y}^{\mathrm{A1}}$ be a solution obtained by our first algorithm. The definition of independent randomized rounding method implies that

$$\begin{aligned}&\mathrm{E}[\max\{y^{\mathrm{A1}}_{t,s}, y^{\mathrm{A1}}_{t,s+1}\}] = \Pr[y^{\mathrm{A1}}_{t,s} = 1 \vee y^{\mathrm{A1}}_{t,s+1} = 1] \\ &= \; 3/4 = (3/2)(1/2) = (3/2)\max\{y^*_{t,s}, y^*_{t,s+1}\}.\end{aligned}$$

From the above, we have

$$\begin{aligned}&\mathrm{E}[Z^{\mathrm{A1}}] = \mathrm{E}[Z^{\mathrm{C}}(\boldsymbol{y}^{\mathrm{A1}})] \\ &= \sum_{t\in T}\left\{\sum_{s\in\{1,\hat{s}\}} \mathrm{E}[y^{\mathrm{A1}}_{t,s}] + \sum_{s\in S\setminus\{\hat{s}\}} \mathrm{E}[\max\{y^{\mathrm{A1}}_{t,s}, y^{\mathrm{A1}}_{t,s+1}\}]\right\} \\ &\leq \sum_{t\in T}\left\{\sum_{s\in\{1,\hat{s}\}} y^*_{t,s} + \sum_{s\in S\setminus\{\hat{s}\}} \frac{3}{2}\max\{y^*_{t,s}, y^*_{t,s+1}\}\right\} \\ &\leq (3/2)Z^{\mathrm{C}}(\boldsymbol{y}^*) = (3/2)Z^{\mathrm{LP}}\end{aligned}$$

Case b-2: Next, we consider our second algorithm. Let $\boldsymbol{y}^{\mathrm{A2}}$ be a solution obtained by our second algorithm. With probability 1/2, $a^*_{t,s} = a^*_{t,s+1}$ and $y^{\mathrm{A2}}_{t,s} = y^{\mathrm{A2}}_{t,s+1}$ hold. In this case, we have $\max\{y^{\mathrm{A2}}_{t,s}, y^{\mathrm{A2}}_{t,s+1}\} = y^{\mathrm{A2}}_{t,s}$. With probability 1/2, $a^*_{t,s} \neq a^*_{t,s+1}$ and $y^{\mathrm{A2}}_{t,s} \neq y^{\mathrm{A2}}_{t,s+1}$ hold. In this case, $\max\{y^{\mathrm{A2}}_{t,s}, y^{\mathrm{A2}}_{t,s+1}\}$ is always equal to 1. Thus, we have

$$\begin{aligned}&\mathrm{E}[\max\{y^{\mathrm{A2}}_{t,s}, y^{\mathrm{A2}}_{t,s+1}\}] = (1/2)(1/2) + (1/2) = 3/4 \\ &= \; (3/2)(1/2) = (3/2)\max\{y^*_{t,s}, y^*_{t,s+1}\}.\end{aligned}$$

From the above we can also show that

$$\mathrm{E}[Z^{\mathrm{A2}}] = \mathrm{E}[Z^{\mathrm{C}}(\boldsymbol{y}^{\mathrm{A2}})] \leq (3/2)Z^{\mathrm{C}}(\boldsymbol{y}^*) = (3/2)Z^{\mathrm{LP}}$$

in the same way as Case b-1.

Case b-3: Lastly, we discuss our third algorithm. Let $\boldsymbol{y}^{\mathrm{A3}}$ be a solution obtained by our third algorithm. The 'flipping' procedure implies the following. With probability 3/4, $a^*_{t,s} = a^*_{t,s+1}$ and $y^{\mathrm{A3}}_{t,s} = y^{\mathrm{A3}}_{t,s+1}$ hold. With probability 1/4, $a^*_{t,s} \neq a^*_{t,s+1}$ and $y^{\mathrm{A3}}_{t,s} \neq y^{\mathrm{A3}}_{t,s+1}$ hold. Thus, we have

$$\begin{aligned}&\mathrm{E}[\max\{y^{\mathrm{A3}}_{t,s}, y^{\mathrm{A3}}_{t,s+1}\}] = (3/4)(1/2) + (1/4) = 5/8 \\ &= \; (5/4)(1/2) = (5/4)\max\{y^*_{t,s}, y^*_{t,s+1}\}.\end{aligned}$$

From the above we can also show that

$$\begin{aligned}
\mathrm{E}[Z^{\mathrm{A3}}] &= \mathrm{E}[Z^{\mathrm{C}}(\boldsymbol{y}^{\mathrm{A3}})] \\
&\leq \sum_{t\in T}\left\{\sum_{s\in\{1,\hat{s}\}} y^*_{t,s} + \sum_{s\in S\setminus\{\hat{s}\}} \frac{5}{4}\max\{y^*_{t,s}, y^*_{t,s+1}\}\right\} \\
&\leq (5/4)Z^{\mathrm{C}}(\boldsymbol{y}^*) = (5/4)Z^{\mathrm{LP}}.
\end{aligned}$$

□

In [2], Bertsimas et al. proposed $(2-(1/2)^k)$-approximation algorithm for MIN kSAT. Thus the second inequalities in the above theorem is a special case of their result. We simplified their proof by restricting $k=2$.

5 Computational Experiments

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

Each of Tables 1, 2 and 3 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 [6]. We used the distance matrix of TSP instance `att48` from TSPLIB [18]. We chose cities of `att48` with indices from 1 to $2n$. For each instance, we applied the algorithms described in Section 3 and generated HA-assignments *upp_itr* times, where $\mathit{upp\_itr} = \min\{\max\{2^{n+1}, 1000\}, 10000\}$. Finally, we output a solution with the best of generated *upp_itr* solutions. In order to evaluate the quality of the best solutions, we solved the same instances with integer programs in a similar manner as Trick [17]. All computations were performed on Dell Dimension 8250 (CPU: Pentium 4, 3.06 GHz, RAM: 512 MB, OS: Vine Linux 2.6) with CPLEX 8.0 [10] for integer programs, XPRESS-MP Workstation (Model Builder 10.04, Integer Optimiser 10.27) [3] for **LP**.

We did not solved integer programs for $2n = 20$ to 40 in the constant case because it would not terminate within reasonable computational time. In Table 1 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.

For the 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 a solution whose ratio is better than that of others (including SDP based approach in [16]) for any number of teams;
(2) all of our procedures based on LP relaxation give more acceptable ratios even by the little difference compared with SDP based approach in [16].

For the constant case, when $2n = 16, 18$, almost all of the averages of approximation ratios are less than 1.20. Contrary to the weighted case, the effectiveness of our third algorithm is emphasized. However, the SDP based approach gives solutions with higher qualities.

As we showed in Theorem 1, **LP** has an optimal solution satisfying half integrality. In Table 1, **half int.** shows the ratios of the number of variables whose value is 1/2. In the weighted case, almost all variables are either 0 or 1, while all variables are 1/2 in the constant case.

For the CPU time in Table 2 and 3, our algorithms are much faster than the SDP based approach proposed in [16] and integer programs. For instance, in the weighted case of $2n = 16$, SDP based approach and integer programs took more than 21 seconds and 65 seconds in average, respectively, while our algorithms take less than 1 second. Moreover, our algorithms terminated less than 8 seconds for any number of teams in the weighted case, and also terminated less than 13 seconds in the constant case. From the above results, it can be said that our algorithms are highly efficient.

6 Conclusions

We proposed a formulation of the home-away assignment problem as an integer program, and performed computational experiments with dependent randomized rounding algorithm based on the Bertsimas, Teo and Vohra's algorithm. Computational experiments showed that our approach is highly effective in terms of quality of solutions and computational speed, in particular, for larger instances.

A Proof of Half Integrality

Here we describe a proof of Theorem 1.

Let **LP** be a linear relaxation of the problem **IP**, described as follows:

(LP)

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

where $w_{t,s}$ $((t,s)\in T\times S\setminus\{\hat{s}\})$ are free continuous variables.

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:

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

where $t' = \tau(t,s)$ and $t'' = \tau(t,s+1)$. Here we note 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 $(\boldsymbol{y}^+,\boldsymbol{w}^+)$, $(\boldsymbol{y}^-,\boldsymbol{w}^-)$ as follows:

for each $(t,s) \in T \times S$, we set

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

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

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

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:

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

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**. Therefore we obtain Theorem 1.

B Generating an HA-assignment

Here we describe an algorithm 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 assigning procedure to the rest of teams. For $s = \hat{s}$, 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 satisfying 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 $[\forall t \in T,\ \forall s \in \{1, 2, \ldots, n-1\},\ a'_{t,2s} = a'_{t,2s+1}]$.

References

[1] Adi Avidor, A., and 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. and 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] Easton, K., Nemhauser, G. and Trick, M.: 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.

[5] Easton, K., Nemhauser, G. and Trick, M.: Sports Scheduling, In: Handbook of Scheduling: Algorithms, Models, and Performance Analysis, Leung, J. Y-T. and Anderson, J. H. (Eds.), Chapman & Hall, 2004.

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

[7] Garey, M. R. and Johnson, D. S.: Computers and Intractability; a Guide to the Theory of NP-completeness, W. H. Freeman, New York, 1979.

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

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

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

[11] Miyashiro, R. and Matsui, T.: Semidefinite programming based approaches to the break minimization problem, Computers and Operations Research (to appear).

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

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

[14] 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.

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

[16] Suzuka, A., Miyashiro, R., Yoshise, A. and 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.

[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] TSPLIB web page, available at the URL
`http://www.crpc.rice.edu/softlib/tsplib/`

Table 1: Results of computational experiments (Approximation Ratio)

	weighted case					
	LP		A1	A2	A3	SDP
$2n$	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

	constant case					
	LP		A1	A2	A3	SDP
$2n$	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 2: Results of computational experiments (CPU time [s]: weighted case)

	A1		A2		A3		SDP		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 3: Results of computational experiments (CPU time [s]: constant case)

	A1		A2		A3		SDP		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 a 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: our first algorithm; **A2**: our second algorithm; **A3**: our third algorithm;
SDP: SDP based approach proposed in [16];
IP : the integer program in a similar manner as Trick [17];
avg.: average; **s. d.**: standard deviation.