INSTITUTE OF POLICY AND PLANNING SCIENCES

Discussion Paper Series

No. 1104

Semidefinite Programming Based Approaches to Home-away Assignment Problems in Sports Scheduling

Ayami SUZUKA Ryuhei MIYASHIRO
Akiko YOSHISE Tomomi MATSUI

February, 2005

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

Semidefinite Programming Based Approaches to Home-away Assignment Problems 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/

February 22, 2005

Abstract

For a given schedule of a round-robin tournament and a matrix of distances between homes of teams, an optimal home-away assignment problem is to find a home-away assignment that minimizes the total traveling distance. We propose a technique to transform the problem to MIN RES CUT. We apply Goemans and Williamson's 0.878-approximation algorithm for MAX RES CUT, which is based on a positive semidefinite programming relaxation, to the obtained MIN RES CUT instances. Computational experiments show that our approach quickly generates solutions of good approximation ratios.

Keywords: sports timetabling; semidefinite programming; Goemans and Williamson's approximation algorithm.

*This paper is also referred as A. Suzuka, R. Miyashiro, A. Yoshise and T. Matsui, "Semidefinite Programming Based Approaches to Home-away Assignment Problems in Sports Scheduling," METR2005-07, Department of Mathematical Engineering and Information Physics, Faculty of Engineering, The University of Tokyo, 2005.

1 Home-away Assignment Problem

Recently, sports scheduling becomes one of the main topics in the area of scheduling (e.g., see "*Handbook of Scheduling*" Chapter 52 (Sports Scheduling) [2]). This paper deals with a home-away assignment problem that assigns home or away to each match of a (double) round-robin tournament so as to minimize the total traveling distance. We propose a technique to transform the problem to MIN RES CUT. We apply Goemans and Williamson's approximation algorithm for MAX RES CUT and report the results of computational experiments.

In the following, we introduce a mathematical definition of the problem. Throughout this paper, we deal with a (double) round-robin tournament with the following properties:

- the number of teams (or players etc.) is $2n$, where $n \in \mathbb{N}$;
- the number of *slots*, i.e., the days when matches are held, is $2(2n-1)$;
- 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 "twice";
- each team plays at the home of every other team exactly "once".

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

A *timetable* is a matrix whose rows are indexed by a set of teams $T = \{1, 2, \ldots, 2n\}$ and columns are indexed by a set of slots $S = \{1, 2, \ldots, 4n - 2\}$. Each entry of a timetable, say $\tau(t, s)$ ($(t, s) \in T \times S$), shows the opponent of team t in slot s. 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 twice;
- 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 slots 7 and 9, and the match in slot 7 is held at the home of team 4, while the other is held at the home of team 2.

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

$T\backslash S$	1	2	3	4	5	6	7	8	9	10
1	3	3	4	4	6	2	5	2	6	5
2	5	5	6	3	3	1	4	1	4	6
3	1	1	5	2	2	4	6	6	5	4
4	6	6	1	1	5	3	2	5	2	3
5	2	2	3	6	4	6	1	4	3	1
6	4	4	2	5	1	5	3	3	1	2

$T\backslash S$	1	2	3	4	5	6	7	8	9	10
1	A	H	A	H	A	A	A	H	H	H
2	H	A	A	A	H	H	A	A	H	H
3	H	A	A	H	A	H	H	A	H	A
4	H	A	H	A	A	A	H	H	A	H
5	A	H	H	H	H	A	H	A	A	A
6	A	H	H	A	H	H	A	H	A	A

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

Given a timetable $\mathcal{T}$, an HA-assignment $\mathcal{A} = (a_{t,s})$ $((t,s) \in T \times S)$ is said to be *consistent* with $\mathcal{T}$ if the followings are satisfied: (C1) $\forall (t,s) \in T \times S$, $\{a_{t,s}, a_{\tau(t,s),s}\} = \{\mathrm{A}, \mathrm{H}\}$, and (C2) $\forall t \in T$, $[\tau(t,s) = \tau(t,s')$ and $s \neq s']$ implies $\{a_{t,s}, a_{t,s'}\} = \{\mathrm{A}, \mathrm{H}\}$ (Condition (C2) is assumed in an ordinary "double" round-robin tournament). A schedule of a round-robin tournament is described as a pair of a timetable and an HA-assignment consistent with the timetable.

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 t to that of t'. We do not assume the symmetricity of $\mathcal{D}$ nor that the distance matrix 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 the HA-assignment, and returns to the home. The *total traveling distance* is the sum total of traveling distances of all the 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 [1, 11]. In this context, the home-away assignment problem is introduced as follows.

HA assignment Problem

Instance: a timetable $\mathcal{T}$ and a distance matrix $\mathcal{D}$.

Task: find an HA-assignment that is consistent with $\mathcal{T}$ and minimizes the total traveling distance.

We formulate the HA assignment problem as MIN RES CUT, and apply Goemans and Williamson's approximation algorithm [5], which is based on the semidefinite programming relaxation. Computational experiments show that our method quickly generates feasible solutions close to optimal.

The rest of this paper is organized as follows: Section 2 proposes formulations of the HA assignment problem as MIN RES CUT; Section 3 reports the results of computational experiments; Section 4 states conclusions.

The problem to find an HA-assignment that is consistent with a given timetable and minimizes the number of breaks (consecutive pairs of home-games) is called the break minimization problem. There are several previous results on this problem (see [10, 12, 3, 7] for example). In [7], Miyashiro and Matsui formulated the break minimization problem as MAX RES CUT and applied Goemans and Williamson's algorithm for MAX RES CUT. Our algorithm proposed in this paper is an extension of their procedure to HA assignment problems. However, we need a non-trivial technique, described in the next section, to extend their procedure to HA assignment problems.

2 Formulation as MIN RES CUT

We propose a formulation of the HA assignment problem as MIN RES CUT. First, we define the problem MIN RES CUT. Let $G = (V, E)$ be an undirected graph with a vertex set V and an edge set E. For any vertex subset $V' \subseteq V$, we define $\delta(V') = \{\{v_i, v_j\} : v_i, v_j \in V,\ v_i \notin V' \ni v_j\}$. The problem MIN RES CUT is defined as follows: given a graph $G = (V, E)$, a specified vertex $r \in V$, a weight function $w : E \longrightarrow \mathbb{R}$, and a set $E_{\mathrm{cut}} \subseteq \{X \subseteq V : |X| = 2\}$, find a vertex subset V' that minimizes $\sum_{e \in \delta(V') \cap E} w(e)$ 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 redundunt for the definition of MIN RES CUT, because for any $V'' \subseteq V$, $\delta(V'') = \delta(V \setminus V'')$. The condition helps to formulate the HA assignment problem as MIN RES CUT. It is easy to show that MIN RES CUT is NP-hard even if $\forall e \in E$, $w(e) = 1$ holds. The problem MAX RES CUT is the maximization version of MIN RES CUT, and Goemans and Williamson [5] proposed a 0.878-approximation algorithm for MAX RES CUT. Now we formulate the HA assignment problem as MIN RES CUT. Given a timetable $\mathcal{T} = (\tau(t, s))$ $((t, s) \in T \times S)$, let $G = (V, E)$ be an undirected graph with a vertex set V and an edge set E defined below.

We introduce an artificial vertex r and define $V = \{v_{t,s} : (t,s) \in T \times S\} \cup \{r\}$, $E = \{\{v_{t,s-1}, v_{t,s}\} : t \in T,\ s \in S \setminus \{1\}\} \cup \{\{r, v_{t,s}\} : (t,s) \in T \times S\}$, and

$$\begin{aligned} E_{\text{cut}} \quad = \quad & \{\{v_{t,s}, v_{\tau(t,s),s}\} : (t,s) \in T \times S\} \\ & \cup \{\{v_{t,s}, v_{t,s'}\} : t \in T,\ s, s' \in S,\ \tau(t,s) = \tau(t,s'), s \neq s'\}. \end{aligned}$$

For a feasible solution V' of this MIN RES CUT instance, i.e., a vertex subset $V' \subseteq V$ satisfying $r \notin V'$ and $E_{\text{cut}} \subseteq \delta(V')$, construct an HA-assignment $\mathcal{A} = (a_{t,s})$ $((t,s) \in T \times S)$ as follows: if $v_{t,s} \in V'$ then $a_{t,s} = \text{A}$, else $a_{t,s} = \text{H}$. This HA-assignment is consistent with $\mathcal{T}$ because (C1) each pair of vertices corresponding to a match is in E_{cut}, and (C2) for each team, every pair of vertices corresponding to matches with a common opponent is in E_{cut}. Obviously, for any consistent HA-assignment, there exists a unique corresponding feasible solution of the MIN RES CUT instance. Thus, there exists a bijection between the feasible set of MIN RES CUT and the set of consistent HA-assignments.

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

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

In the following, we use the notations $t' = \tau(t,s)$ and $t'' = \tau(t,s+1)$ for simplicity. We show that the traveling distance $\ell(t,s)$ satisfy the following equations;

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

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

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

and

$$\begin{aligned}&|v_{t,s} \cap V'|\,|v_{t,s+1} \cap V'| \\ &= \frac{|\{v_{t,s}, r\} \cap \delta(V')| + |\{v_{t,s+1}, r\} \cap \delta(V')| - |\{v_{t,s}, v_{t,s+1}\} \cap \delta(V')|}{2}.\end{aligned} \tag{2}$$

Equations (1) and (2) are obtained from the properties that $r \notin V'$ and $\forall V' \subseteq V,\ |\delta(V') \cap \{\{r, v_{t,s}\}, \{r, v_{t,s+1}\}, \{v_{t,s}, v_{t,s+1}\}\}| \in \{0, 2\}$. The third equality is trivial. Here we note that, if we employ only Equations (1), $\ell(t, s)$ becomes a quadratic function of $|\{v_{t,s}, r\} \cap \delta(V')|$ and $|\{v_{t,s+1}, r\} \cap \delta(V')|$. Using Equation (2), we can transform the quadratic function to a linear function of $|\{v_{t,s}, r\} \cap \delta(V')|$, $|\{v_{t,s+1}, r\} \cap \delta(V')|$ and $|\{v_{t,s}, v_{t,s+1}\} \cap \delta(V')|$.

In a similar way, we can show that the traveling distance of team t before the first slot and after the last slot, denoted by $\ell(t, 0)$ and $\ell(t, 4n-2)$ respectively, satisfy that

$$\begin{aligned}\ell(t, 0) &= d(t, \tau(t, 1))|\{v_{t,1}, r\} \cap \delta(V')|, \\ \ell(t, 4n-2) &= d(\tau(t, 4n-2), t)|\{v_{t,4n-2}, r\} \cap \delta(V')|.\end{aligned}$$

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

$$\begin{aligned}\sum_{t \in T} \sum_{s=0}^{4n-2} \ell(t, s) &= \sum_{t \in T} \sum_{s=1}^{4n-3} \begin{pmatrix} \frac{d(t', t'') - d(t, t'') + d(t', t)}{2} |\{v_{t,s}, r\} \cap \delta(V')| \\ + \frac{d(t', t'') + d(t, t'') - d(t', t)}{2} |\{v_{t,s+1}, r\} \cap \delta(V')| \\ + \frac{-d(t', t'') + d(t, t'') + d(t', t)}{2} |\{v_{t,s}, v_{t,s+1}\} \cap \delta(V')| \end{pmatrix} \\ &\quad + \sum_{t \in T} d(t, \tau(t, 1))|\{v_{t,1}, r\} \cap \delta(V')| \\ &\quad + \sum_{t \in T} d(\tau(t, 4n-2), t)|\{v_{t,4n-2}, r\} \cap \delta(V')|.\end{aligned}$$

Thus, by introducing an appropriate weight function $w : E \to \mathbb{R}_+$ (a precise description appears in Appendix), the total traveling distance satisfies that

$$\sum_{t \in T} \sum_{s=0}^{4n-2} \ell(t, s) = \sum_{e \in E} w(e)|e \cap \delta(V')| = \sum_{e \in E \cap \delta(V')} w(e)$$

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

Here we note that the break maximization problem, which maximizes the number of consecutive pairs of home-games, is a special case of our problem such that the distance between any pair of homes is equal to 1. It is shown in [8] that the break maximization problem is essentially equivalent to the break minimization problem, which is discussed in many papers [10, 12, 3, 7]. Thus, the break minimization problem is also a special case of the HA assignment problem discussed in this paper.

3 Computational Experiments

For MAX RES CUT, Goemans and Williamson [5] proposed a 0.878-randomized approximation algorithm using semidefinite programming. Here we apply Goemans and Williamson's algorithm to the proposed MIN RES CUT formulation of the HA assignment problem. In the following, we briefly explain the procedure. The algorithm consists of the following three steps.

1. Semidefinite Programming

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

$$\begin{array}{ll} \text{minimize} & \boldsymbol{C} \bullet \boldsymbol{X} \\ \text{subject to} & \boldsymbol{E}_{ii,ii} \bullet \boldsymbol{X} = 1 \qquad (\forall i \in V), \\ & \boldsymbol{E}_{ij,ji} \bullet \boldsymbol{X} = -2 \quad (\forall \{i,j\} \in E_{\text{cut}}), \\ & \boldsymbol{X} \succeq \boldsymbol{O},\ \boldsymbol{X} \text{ is symmetric, } \boldsymbol{X} \in \mathbb{R}^{V \times V}, \end{array}$$

   where $\boldsymbol{C} = (\text{diag}(\boldsymbol{W}\mathbf{1}) - \boldsymbol{W})/4$, $\boldsymbol{X} \bullet \boldsymbol{Y} = \sum_i \sum_j X_{ij} Y_{ij}$, $\boldsymbol{E}_{ij,ji}$ is the matrix in which entries E_{ij} and E_{ji} are ones and every other entry is zero, and $\boldsymbol{X} \succeq \boldsymbol{O}$ means that $\boldsymbol{X}$ is positive semidefinite.

2. Cholesky Decomposition

   Decompose an (almost) optimal solution $\boldsymbol{X}_0$ of the semidefinite programming problem in Step 1 into a matrix $\widehat{\boldsymbol{X}}$ such that $\boldsymbol{X}_0 = \widehat{\boldsymbol{X}}^\top \widehat{\boldsymbol{X}}$ (Cholesky decomposition).

3. Hyperplane Separation

   Generate a vector $\boldsymbol{u}$ at uniformly random on the surface of d-dimensional unit ball and put $V_1 = \{i \in V : \boldsymbol{u}^\top \widehat{\boldsymbol{x}}_i \geq 0\}$ where d is the number of rows of $\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{x}}_i$ is the column vector of $\widehat{\boldsymbol{X}}$ index by $i \in V$. Output a vertex subset $V' = \begin{cases} V_1 & (\text{if } r \notin V_1), \\ V \setminus V_1 & (\text{if } r \in V_1). \end{cases}$

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

Goemans and Williamson [5] showed that the maximization version of the above algorithm finds a feasible solution of MAX RES CUT, and its expected objective value is at least 0.87856 times the optimal value. In case of MIN RES CUT, any non-trivial bound of approximation ratio of the above algorithm is not known.

Finally, we report our computational results. Computational experiments were performed as follows. Tables 1 (a) and (b) show the results when we generated 10 timetables for each size of $2n = 16, 18, 20, 22, 24, 26, 30$. We constructed a timetable of "double" round robin tournament by concatenating two copies of a timetable of "single" round robin tournament that is randomly created as the method described in [3]. The results are shown in Table 1 (a). Table 1 (b) reports the results when each timetable is obtained by concatenating two mutually different timetables of "single" round robin tournament. We used the distance matrix obtained from TSP instance `att48` from TSPLIB. We chose cities of `att48` with indices from 1 to $2n$.

For each instance, we applied Goemans and Williamson's algorithm and generated 10000 HA-assignments by executing Step 3 of the algorithm 10000 times. Finally, we output a solution with the best of generated 10000 solutions. In order to evaluate the quality of the best solutions, we solved the same instances with integer programming in a similar manner as Trick [12]. All computations were performed on Dell Dimension 8100 (CPU: Pentium4, 1.4GHz, RAM: 768MB, OS: Vine Linux 2.6) with SDPA 6.0 [13] for semidefinite programming problems and CPLEX 8.0 [6] for integer programming problems.

Table 1 shows the results of the experiments. In almost of all cases the average of approximation ratios is less than 1.18. We did not solved 26 and 30 teams instances with integer programming because it would not terminate within reasonable computational time. The computational time for our procedure is less than 670 seconds when $2n \leq 24$.

4 Conclusions

We proposed a formulation of HA assignment problems as MIN RES CUT problems, and performed computational experiments with Goemans and Williamson's algorithm for MAX RES CUT, based on semidefinite programming relaxation. Computational experiments showed that our approach is highly effective in terms of quality of solutions and computational speed, in particular, for a large instance.

Appendix

We define a weight function $w : E \to \mathbb{R}_+$, which is discussed in Section 2, as follows:

$$\begin{aligned}
w(\{v_{t,s}, r\}) &= \frac{d(\tau(t,s),\tau(t,s+1)) - d(t,\tau(t,s+1)) + d(\tau(t,s),t)}{2} \\
&\quad + \frac{d(\tau(t,s-1),\tau(t,s)) + d(t,\tau(t,s)) - d(\tau(t,s-1),t)}{2} \\
&\qquad (\forall t \in T, \forall s \in S \setminus \{1, 4n-2\}), \\
w(\{v_{t,1}, r\}) &= d(t,\tau(t,1)) + \frac{d(\tau(t,1),\tau(t,2)) - d(t,\tau(t,2)) + d(\tau(t,1),t)}{2}, \\
w(\{v_{t,4n-2}, r\}) &= d(\tau(t,4n-2),t) \\
&\quad + \frac{d(\tau(t,4n-3),\tau(t,4n-2)) + d(t,\tau(t,4n-2)) - d(\tau(t,4n-3),t)}{2}, \\
w(\{v_{t,s}, v_{t,s+1}\}) &= \frac{-d(\tau(t,s),\tau(t,s+1)) + d(t,\tau(t,s+1)) + d(\tau(t,s),t)}{2} \\
&\qquad (\forall t \in T, \forall s \in S \setminus \{4n-2\}).
\end{aligned}$$

Then, the total traveling distance satisfies that

$$\begin{aligned}
\sum_{t \in T} \sum_{s=0}^{4n-2} \ell(t,s) &= \sum_{t \in T} \sum_{s=1}^{4n-3} \left(\begin{array}{l} \frac{d(t',t'') - d(t,t'') + d(t',t)}{2} |\{v_{t,s}, r\} \cap \delta(V')| \\ + \frac{d(t',t'') + d(t,t'') - d(t',t)}{2} |\{v_{t,s+1}, r\} \cap \delta(V')| \\ + \frac{-d(t',t'') + d(t,t'') + d(t',t)}{2} |\{v_{t,s}, v_{t,s+1}\} \cap \delta(V')| \end{array} \right) \\
&\quad + \sum_{t \in T} d(t,\tau(t,1)) |\{v_{t,1}, r\} \cap \delta(V')| + \sum_{t \in T} d(\tau(t,4n-2),t) |\{v_{t,4n-2}, r\} \cap \delta(V')| \\
&= \sum_{e \in E} w(e) |e \cap \delta(V')| = \sum_{e \in E \cap \delta(V')} w(e).
\end{aligned}$$

Thus, the objective function value of MIN RES CUT with respect to $w(e)$ defined above is equivalent to the total traveling distance.

References

[1] 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.

[2] 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.

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

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

[5] 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.

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

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

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

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

[10] 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.

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

[12] 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.

[13] Yamashita, M., Fujisawa, K. and Kojima, M.: Implementation and evaluation of SDPA 6.0, Optimization Methods and Software **18** (2003) 491–505.

Table 1: Results of computational experiments

(a)

#teams	ratio	SDP (s)		IP (s)	
	avg.	avg.	(s. d.)	avg.	(s. d.)
16	1.00119	81.8	(2.82)	13.2	(2.09)
18	1.11825	129.1	(23.99)	27.4	(8.87)
20	1.00119	233.1	(13.82)	61.1	(25.68)
22	1.00122	388.4	(15.10)	1550.3	(1124.54)
24	1.00478	617.7	(22.18)	68341.7	(124286.75)
26	—	989.4	(22.68)	—	—
30	—	2142.9	(104.27)	—	—

(b)

#teams	ratio	SDP (s)		IP (s)	
	avg.	avg.	(s. d.)	avg.	(s. d.)
16	1.00057	86.2	(2.82)	22.3	(7.71)
18	1.17323	123.9	(28.84)	61.1	(26.40)
20	1.00071	273.8	(19.26)	328.2	(419.23)
22	1.00099	393.0	(9.49)	1244.5	(748.60)
24	1.00121	664.1	(23.83)	13078.3	(19549.06)
26	—	1057.3	(30.40)	—	—
30	—	2226.3	(78.00)	—	—

#teams: the number of teams;
ratio: average of ratios of the optimal value and the objective function value of the best solutions obtained by our procedure;
SDP : computational time for our procedure;
IP : computational time for integer programming;
avg.: average; s. d.: standard deviation.