MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Semidefinite Programming Based Approaches to the Break Minimization Problem

Ryuhei MIYASHIRO and Tomomi MATSUI

METR 2003–28 August 2003

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm

Semidefinite Programming Based Approaches to the Break Minimization Problem†

Ryuhei MIYASHIRO and Tomomi MATSUI

Department of Mathematical Informatics
Graduate School of Information Science and Technology
The University of Tokyo
{miya,tomomi}@misojiro.t.u-tokyo.ac.jp

August 2003

Abstract

This paper deals with the break minimization problem of sports scheduling. The problem is to find a stadium assignment that minimizes the total number of breaks under a given round-robin tournament schedule. We show that the problem can be formulated as MAX RES CUT and MAX 2SAT. We also apply Goemans and Williamson's approximation algorithm based on positive semidefinite programming relaxation. Our computational experiments show that the approximation algorithm finds good solutions in practical computational time.

Keywords: sports scheduling, round-robin tournament, positive semidefinite programming, approximation algorithm, MAX RES CUT, MAX CUT, MAX 2SAT

1 Introduction

During the last two decades, there has been considerable advance in the area of automated sports timetabling. Schedules of several kinds of sports were discussed in papers [1, 2, 3, 5, 6, 14, 34], and most of the schedules are round-robin tournaments. Recently, many algorithms for constructing round-robin tournaments have been reported, which are based on a variety of optimization techniques: constraint programming [4, 18, 20, 21, 22, 25, 35], simulated annealing [41], tabu search [7, 19, 42], SAT solver [43], and integer programming [15, 30]. The mathematical structure of the scheduling

†Supported by Superrobust Computation Project of the 21st Century COE Program "Information Science and Technology Strategic Core."

problem of round-robin tournaments is also discussed in several papers [8, 9, 10, 11, 27, 28, 32, 33, 36, 37, 39, 40].

Some authors proposed to consider a round-robin tournament schedule without a stadium assignment that obeys various constraints and to find a stadium assignment that minimizes the number of breaks [12, 31, 38]. Such a problem is called the "break minimization problem," which is concerned in this paper. We show that the problem can be formulated as MAX RES CUT and MAX 2SAT, and apply Goemans and Williamson's approximation algorithm [17] based on positive semidefinite programming relaxation. Our computational experiments show that the approximation algorithm finds good solutions in practical computational time.

In the next section, we introduce the break minimization problem in detail. Section 3 describes formulations of the break minimization problem as MAX RES CUT and MAX 2SAT. In Section 4, we report some computational experiments. Finally, we state a conclusion and related open problems in Section 5.

2 Break Minimization Problem

In the rest of this paper, we consider a round-robin tournament consisting of $2n$ teams, and denote the set of teams by $N = \{1, 2, \ldots, 2n\}$ and the set of slots by $S = \{1, 2, \ldots, 2n-1\}$. In a round-robin tournament, each team plays against every other team once. It is a well-known fact that, for any positive integer n, there is at least one round-robin tournament.

We describe a possible round-robin tournament by a matrix T, called a *tournament chart*, whose rows and columns are indexed by N and S, respectively. The cell of T indexed by $(i, s) \in N \times S$, denoted by $t_{is} \in N$, represents the opponent of team i at slot s. Thus, a matrix T corresponds to a round-robin tournament if and only if T satisfies the following:
(1) for each team $i \in N$, the i-th row of T is a permutation of $N \setminus \{i\}$;
(2) for any $(i, s) \in N \times S$, $t_{i's} = i$ where $i' = t_{is}$.
Figure 1(a) shows an example of a tournament chart of 8 teams.

In this paper, we assume that each game is held at the home of one of the teams playing. If the game between i and j is played at the home of i, the game is called a *home-game* for i and an *away-game* for j. We describe a *stadium assignment* $\mathcal{A}$ by a digraph (N, A) satisfying that for any pair of vertices (teams) $\{i, j\}$, exactly one of the two ordered pairs (i, j) and (j, i) is a directed arc in $\mathcal{A}$ (the corresponding concept in graph theory is called a "tournament graph.") When $\mathcal{A}$ contains the arc (i, j), then the game between i and j is held at the home of j. Given a pair of a tournament chart T and a stadium assignment $\mathcal{A}$, we say that the cell (i, s) is a home-game (away-game) if the game between i and t_{is} is held at the home of i (t_{is}).

	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

(a)

	1	2	3	4	5	6	7
1	A	H	A	H	A	H	A
2	H	A	H	A	A	H	A
3	H	A	H	A	H	H	A
4	A	A	H	A	H	A	H
5	H	A	H	A	H	A	H
6	A	H	A	H	H	A	H
7	A	H	A	H	A	A	H
8	H	H	A	H	A	H	A

(b)

Figure 1: Tournament chart of 8 teams and one of its optimal assignment

In sports scheduling, it is considered undesirable if a team plays two consecutive games either both at home or both at away. If team i has two home-games (two away-games) at slots $s-1$ and s, we say that i has an *HH-break* (*AA-break*) at s. If team i has either HH-break or AA-break at slot s, we simply say that i has a *break* at s. Given a tournament chart T, the *break minimization problem* is to find a stadium assignment that minimizes the total number of breaks. Figure 1(b) shows an optimal assignment, which has six breaks, with respect to the tournament chart of Figure 1(a).
To the best of our knowledge, the complexity status of the break minimization problem remains open. In the recent paper [12], Elf et al. conjectured NP-hardness of the problem.

It is well-known that for any tournament chart T, the number of breaks is greater than or equal to $2n-2$, where $2n$ is the number of teams. In addition, for any positive integer n, there is at least one pair of a tournament chart and a stadium assignment such that the number of breaks attains the lower bound $2n-2$ (see de Werra [8] for example).

For any stadium assignment corresponding to a tournament chart, the number of away-games is equal to the number of home-games in each slot. Thus, it is easy to see that for each slot s, the number of HH-breaks at s is equal to the number of AA-breaks at the same slot. It directly implies that the break minimization problem is essentially equivalent to the problem that minimizes the number of AA-breaks (HH-breaks).

3 Formulations as MAX RES CUT and MAX 2SAT

First, we formulate the break minimization problem as MAX RES CUT. Given a tournament chart T, we introduce two undirected graphs G_1, G_2 as follows. Both G_1 and G_2 have a vertex set $V = N \times S$. The graph G_1 has an edge set $E_1 \stackrel{\text{def.}}{=} \{\{u,v\} : \exists(i,s) \in N \times (S \setminus \{1\}),\ u = (i, s-1),\ v =$

$(i,s)\}$. Each edge in G_1 corresponds to a pair of consecutive cells in a row of the tournament chart. The graph G_2 has an edge set $E_2 \stackrel{\text{def.}}{=} \{\{u,v\} : \exists(i,s) \in N \times S,\ u = (i,s),\ v = (t_{is}, s)\}$. Here we note that each edge in E_2 corresponds to a game. Given a vertex subset $V' \subseteq V$, the set of edges $\delta_i(V') \stackrel{\text{def.}}{=} \{\{u,v\} \in E_i : u \in V', v \in V \setminus V'\}$ is called the *cut set* with respect to V' on the graph G_i $(i \in \{1,2\})$.

Let $\mathcal{A}$ be a stadium assignment and $V' \subseteq V$ be the set of vertices (cells of the given tournament chart T) corresponding to home games with respect to $\mathcal{A}$. It is obvious that $\delta_2(V') = E_2$. Additionally, an edge $\{(i,s-1),(i,s)\} \in E_1$ is not contained in the cut set $\delta_1(V')$ if and only if team i has a break at slot s. Then it is easy to see that the break minimization problem is to find a vertex subset $V' \subseteq V$ that minimizes $|E_1 \setminus \delta_1(V')| = 2n(2n-2) - |\delta_1(V')|$ under the condition that $\delta_2(V') = E_2$. The maximization version of the problem is

$$\text{P1: } \max\{|\delta_1(V')| : V' \subseteq V,\ \delta_2(V') = E_2\}.$$

Problem P1 is a special case of MAX RES CUT discussed by Goemans and Williamson [17]. We can apply their approximation algorithm based on positive semidefinite programming to Problem P1 and its approximation ratio is known to be 0.87856.

The idea of the above formulation also appeared in the paper [12] by Elf et al. In addition, they transformed the problem to MAX CUT with negative edge weights by eliminating the constraint $\delta_2(V') = E_2$. When we apply Goemans and Williamson's algorithm to an instance of MAX CUT with negative edge weights, the proposed approximation ratio 0.87856 is not guaranteed (see the original paper [17] for detail).

Next, we formulate the break minimization problem as a special case of MAX 2SAT. For each vertex $v = (i,s) \in V$, we introduce a Boolean variable x_{is}. Given a tournament chart T and a stadium assignment $\mathcal{A}$, we set $x_{is} = \texttt{TRUE}$ if and only if the cell (i,s) is a home-game with respect to $\mathcal{A}$. Then it is clear that for any edge $\{(i,s),(j,s)\} \in E_2$, the equation $\neg x_{is} = x_{js}$ holds. Additionally, team i has an AA-break at slot s if and only if the clause $x_{i(s-1)} \vee x_{is}$ is $\texttt{FALSE}$. As described in the previous section, the break minimization problem is essentially equivalent to the problem that minimizes the number of AA-beaks. Thus, the break minimization problem is to find a true-false assignment that minimizes the number of unsatisfied clauses in the set $\{(x_{i(s-1)} \vee x_{is}) : \{(i,s-1),(i,s)\} \in E_1\}$ under the conditions that $\neg x_{is} = x_{it}$ $(\forall \{(i,s),(i,t)\} \in E_2)$. The maximization version of the above problem is

$$\text{P2: } \max\left\{ |E'| : \begin{array}{l} E' \subseteq E_1,\ x_u \vee x_v = \texttt{TRUE}\ (\forall \{u,v\} \in E'), \\ \neg x_u = x_v\ (\forall \{u,v\} \in E_2),\ x_u \in \{\texttt{TRUE}, \texttt{FALSE}\}\ (\forall u \in V) \end{array} \right\}.$$

Table 1: Approximation ratios for MAX 2SAT

ratio	authors	year	paper
0.87856	Goemans and Williamson	1995	[17]
0.931	Feige and Goemans	1995	[13]
0.935	Matuura and Matsui	2003	[26]
0.940	Lewin, Livnat and Zwick	2002	[24]

By eliminating one variable for each constraint $\neg x_u = x_v$ in Problem P2, we obtain an instance of MAX 2SAT. Goemans and Williamson [17] also proposed an approximation algorithm for MAX 2SAT and its approximation ratio is 0.87856. Table 1 shows a list of papers on approximation algorithms for MAX 2SAT.

It is not hard to see that when we apply the approximation algorithm for MAX 2SAT proposed by Goemans and Williamson to Problem P2, the practical procedure is equivalent to that obtained by applying the approximation algorithm for MAX RES CUT proposed by Goemans and Williamson to Problem P1. In the next section, we show some results of computational experiments with Goemans and Williamson's algorithm. We do not implement the other algorithms proposed in the papers [13, 26, 24] because it is impractical in our case. For example, when the number of teams is greater than or equal to 12, the number of constraints becomes more than one million. More precisely, with the algorithms in [13, 26, 24], the number of variables and that of constraints are $O(n^4)$ and $O(n^6)$, respectively.

4 Computational Experiments

In this section, we report the results of computational experiments that show the efficiency of our procedure. In the following, we regard the break minimization problem as the maximization problem instead. Accordingly, the objective function of the problem to be maximized is the number of "non-break edges in E_1," not that of breaks.

First, we randomly created one tournament chart for each size of $2n = 16, 18, \ldots, 26, 30, 40$. Then, we applied Goemans and Williamson's algorithm and generated 2000 stadium assignments for each tournament chart by executing the hyperplane separation procedure 2000 times (see [17] for detail). Here we note that Goemans and Williamson's algorithm is a Monte-Carlo type randomized algorithm and therefore generated solutions depend on an employed random number generator. All computations were performed on SUN Ultraspark 5.4 Workstation (CPU: 900 MHz, RAM: 2 GB). We used SDPA 5.0 [16] to solve positive semidefinite programming prob-

Table 2: Results of Experiments

#teams	#non-breaks	diff.	average	time (s)	IP time (s)
16	192	0	0.967	3.1	34
18	248	0	0.955	6.0	207
20	312	2	0.951	10.4	1346
22	382	2	0.954	17.6	7802
24	458	4	0.955	30.6	109024
26	536	4–6*	0.949<	49.6	259200<
30	720	—	—	117	N/A
40	1306	—	—	670	N/A

#teams: the number of teams;
#non-breaks: the objective value of the best solution obtained by our procedure;
diff.: the difference between #non-breaks and the optimal value obtained by integer programming;
average: the average of approximation ratios of generated 2000 solutions (should be larger than 0.878);
time: computational time;
IP time: computational time for solving the integer programming problem.
***Note:** In the 26 teams instance, integer programming did not terminate within three days. The best objective value and upper bound obtained are 540 and 543.8, respectively. For larger instances, computations of integer programming did not terminate within reasonable computational times.

lems, and implemented the code for the hyperplane separation procedure in C. In order to evaluate the quality of the obtained solutions, we solved the same maximization problems by using integer programming with ILOG CPLEX 7.0 [23]. We formulated the maximizing problems as integer programming problems in the same manner as Trick's paper [38].

Table 2 shows that our approach produced very good solutions quickly. In 16 and 18 team instances, optimal solutions are obtained. In 20 and 22 teams instances, the differences are only two. (Note that the number of non-breaks is always even.) In the 26 teams instance, integer programming did not terminate within 3 days. The best objective value and upper bound obtained by integer programming are 540 and 543.8, respectively. Thus, the difference from the objective value we obtained is 4 or 6. If the optimal value is 540 (542), the average is 0.9527 (0.9492). The averages of approximation ratios of generated solutions, which should be theoretically larger than 0.878, is about 0.95 in every instance (when it was solved by integer programming). The computational time of each instance is quite short. Furthermore, our method can be applied to larger instances, such as 40 teams, which has much larger size ever considered in papers [12, 31, 38]. It should be noted that their emphasis is placed on getting an optimal solution, whereas our aim is to generate better solutions quickly.

5 Conclusion

We proposed positive semidefinite programming based approaches to the break minimization problem. Our computational experiments showed that Goemans and Williamson's algorithm is highly effective in terms of both quality of the solutions and computational time.

There are some open problems on the break minimization problems.

1. What is the computational complexity of the break minimization problem?
2. Is there any correspondence between computational time to obtain an optimal solution and the optimal value?

For the first question, Elf et al. [12] conjectured that this problem is NP-hard. To the best of our knowledge, this problem is still open. For the second one, Elf et. al [12] reported that, with both their algorithm and Trick's integer programming approach [38], computational times decrease when the optimal number of breaks becomes smaller. Recently, we proposed a polynomial time algorithm to determine whether the optimal value of a given instance of the break minimization problem is less than the number of teams [29].

References

[1] J. R. Armstrong and R. J. Willis, "Scheduling the cricket world cup — a case study," Journal of the Operations Research Society, 44 (1993), pp. 1067–1072.

[2] B. C. Ball and D. B. Webster, "Optimal scheduling for even-numbered team athletic conferences," AIIE Transactions, 9 (1977), pp. 161–169.

[3] J. C. Bean and J. R. Birge, "Reducing traveling costs and player fatigue in the national basketball association," Interface, 10 (1980), pp. 98–102.

[4] T. Benoist, F. Laburthe, and B. Rottembourg, "Lagrange relaxation and constraint programming collaborative schemes for traveling tournament problems," Proc. of International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems, pp. 15–26, 2001.

[5] D. C. Blest and D. G. Fitzgerald, "Scheduling sports competitions with a given distribution of times," Discrete Applied Mathematics, 22 (1988/1989), pp. 9–19.

[6] W. O. Cain Jr., "The computer-assisted heuristic approach used to schedule the major league baseball clubs," in S. P. Ladany and R. E. Machol (eds.), Optimal Strategies in Sports, Studies in Management Science and Systems, North-Holland, Amsterdam, pp. 32–41, 1977.

[7] D. Costa, "An evolutionary tabu search algorithm and the NHL scheduling problem," INFOR, 33 (1995), pp. 161–178.

[8] D. de Werra, "Geography, games and graphs," Discrete Applied Mathematics, 2 (1980), pp. 327–337.

[9] D. de Werra, "Minimizing irregularities in sports schedules using graph theory," Discrete Applied Mathematics, 4 (1982), pp. 217–226.

[10] D. de Werra, "On the multiplication of divisions: the use of graphs for sports scheduling," Networks, 15 (1985), pp. 125–136.

[11] D. de Werra, "Some models of graphs for scheduling sports competitions," Discrete Applied Mathematics, 21 (1988), pp. 47–65.

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

[13] U. Feige and M. X. Goemans, "Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT," Proc. of 3rd Israel Symposium on the Theory of Computing and Systems, pp. 182–189, 1995.

[14] J. A. Ferland and C. Fleurent, "Computer aided scheduling for a sport league," INFOR, 29 (1991), pp. 14–25.

[15] J. A. Ferland and C. Fleurent, "Allocating games for the NHL using integer programming," Operations Research, 41 (1993), pp. 649–654.

[16] K. Fujisawa, M. Kojima, K. Nakata, and M. Yamashita, "SDPA User's manual — version 5.00," Research Report on Mathematical and Computing Sciences Series B: Operations Research, Department of Mathematical and Computing Science, Tokyo Institute of Technology, 1999 (available at http://www.is.titech.ac.jp/~yamashi9/sdpa/).

[17] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems Using Semidefinite Programming," Journal of the ACM, 42 (1995), pp. 1115–1145.

[18] C. P. Gomes, B. Selman, and H. Kautz, "Boosting combinatorial search through randomization," Proc. of AAAI-98, Madison/WI, USA, pp. 431–437, 1998.

[19] J. P. Hamiez and J. K. Hao, "Solving the sports league scheduling problem with tabu search," Lecture Notes in Artificial Intelligence, 2148 (2001), Springer-Verlag, pp. 24–36.

[20] M. Henz, "Constraint-based round robin tournament planning," Proc. of ICLP, pp. 545–557, 1999.

[21] M. Henz, "Scheduling a major college basketball conference — revisited," Operations Research, 49 (2001), pp. 163–168.

[22] M. Henz, T. Müller, and S. Thiel, "Global constraints for round robin tournament scheduling," European Journal of Operational Research, to appear.

[23] ILOG, "ILOG CPLEX 7.0 User's Manual," ILOG, Gentilly, 2000.

[24] M. Lewin, D. Livnat, and U. Zwick, "Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems." Proc. of IPCO, Lecture Notes in Computer Science, 2337 (2002), Springer-Verlag, pp. 67–82.

[25] K. McAloon, C. Tretkoff, and G. Wetzel, "Sports league scheduling," Proc. of ILOG Optimization Suite International Users' Conference, 1997.

[26] S. Matuura and T. Matsui, "New approximation algorithms for MAX 2SAT and MAX DICUT," Journal of the Operations Research Society of Japan, 46 (2003), pp. 178–188.

[27] R. Miyashiro and T. Matsui, "Notes on equitable round-robin tournaments," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E85-A (2002), pp. 1006–1010.

[28] R. Miyashiro, H. Iwasaki, and T. Matsui, "Characterizing feasible pattern sets with a minimum number of breaks," Practice and Theory of Automated Timetabling IV, Lecture Notes in Computer Science, 2740 (2003), Springer-Verlag, to appear.

[29] R. Miyashiro and T. Matsui, "Round-Robin Tournaments with a Small Number of Breaks," Mathematical Engineering Technical Reports, METR 2003-29, Department of Mathematical Informatics, Graduate School of Information Science and Technology, the University of Tokyo, 2003 (available at http://www.keisu.t.u-tokyo.ac.jp/Research/techrep.0.html).

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

[31] J. C. Régin, "Minimization of the number of breaks in sports scheduling problems using constraint programming," in E. C. Freuder and R. J. Wallace (eds.), Constraint Programming and Large Scale Discrete Optimization, DIMACS, 57 (1998), American Mathematical Society Publications, pp. 115–130.

[32] A. Rosa and W. D. Wallis, "Premature sets of 1-factors or how not to schedule round robin tournaments," Discrete Applied Mathematics, 4 (1982), pp. 291–297.

[33] K. G. Russell, "Balancing carry-over effects in round robin tournaments," Biometrika, 67 (1980), pp. 127–131.

[34] R. A. Russel and J. M. Y. Leung, "Devising a cost effective schedule for a baseball league," Operations Research, 42 (1994), pp. 614–625.

[35] A. Schaerf, "Scheduling sport tournaments using constraint logic programming," Constraints, 4 (1999), pp. 43–65.

[36] J. A. M. Schreuder, "Constructing timetables for sport competitions," Mathematical Programming Study, 13 (1980), pp. 58–67.

[37] J. A. M. Schreuder, "Combinatorial aspects of construction of competition Dutch professional football leagues," Discrete Applied Mathematics, 35 (1992), pp. 301–312.

[38] M. A. Trick, "A schedule-then-break approach to sports timetabling," Practice and Theory of Automated Timetabling III, Lecture Notes in Computer Science, 2079 (2000), Springer-Verlag, pp. 242–253.

[39] A. van Weert and J. A. M. Schreuder, "Construction of basic match schedules for sports competitions by using graph theory," Practice and Theory of Automated Timetabling II, Lecture Notes in Computer Science, 1408 (1998), Springer-Verlag, pp. 201–210.

[40] W. D. Wallis, "A tournament problem," Journal of Australian Mathematical Society Ser. B, 24 (1982), pp. 289–291.

[41] R. J. Willis and B. J. Terrill, "Scheduling the Australian state cricket season using simulated annealing," Journal of the Operations Research Society, 45 (1994), pp. 276–280.

[42] M. Wright, "Timetabling county cricket fixtures using a form of tabu search," Journal of the Operations Research Society, 45 (1994), pp. 758–770.

[43] H. Zhang, "Generating college conference basketball schedules by a SAT solver," Proc. of 5th International Symposium on the Theory and Applications of Satisfiability Testing, pp. 281–291, 2002.