Minimizing the Carry-Over Effects Value in a Round-Robin Tournament

Ryuhei Miyashiro[1] and Tomomi Matsui[2]

[1] Institute of Symbiotic Science and Technology,
Tokyo University of Agriculture and Technology,
Naka-cho, Koganei, Tokyo 184-8588, Japan
r-miya@cc.tuat.ac.jp
[2] Department of Information and System Engineering,
Faculty of Science and Engineering, Chuo University,
Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan
matsui@ise.chuo-u.ac.jp

In this abstract, we deal with the problem of minimizing the carry-over effects value in a round-robin tournament. The carry-over effects value is an index of quality of a round-robin tournament schedule. We propose an effective algorithm for generating schedules of small carry-over effects values. The proposed algorithm produces better schedules than the previous best ones in short computational time.

A round-robin tournament with the following properties is considered in this abstract:
- the number of teams (or players etc.), n, is even;
- the number of *slots*, i.e., the days when matches are held, is $n-1$;
- each team plays one match in each slot;
- each team plays every other team once.

Suppose that, in a round-robin tournament of high-contact sports (such as rugby and American football), team 2 is very strong and another team will be exhausted after the match against team 2. In this situation, which is a better schedule, Figs. 1 or 2? In Fig. 1, five of seven opponents of team 1 play team 1 just after playing team 2. Accordingly, team 1 is considered to have much advantage due to team 2. On the other hand, in Fig. 2 each team (except team 2) derives the advantage from team 2 at most once. In this regard, the schedule of Fig. 2 is better than that of Fig. 1. Such quality of a schedule can be measured by the *carry-over effects value*. In the following, the definition of the carry-over effects value is introduced.

It is said that team i gives a *carry-over effect* to team j if a team plays i in slot s then j in slot $s+1$ ($s \in \{1, 2, \ldots, n-1\}$; regard slot n as slot 1). For a given schedule, the *carry-over effects matrix* C (coe-matrix for short) is a non-negative matrix whose element c_{ij} denotes the number of carry-over effects given by team i to team j in the schedule. By its definition, every coe-matrix satisfies the following:
- the sum of each row is $n-1$;
- the sum of each column is $n-1$;
- every diagonal element is 0.

E. K. Burke, H. Rudová (Eds.): PATAT 2006, pp. 460–463. ISBN 80-210-3726-1.

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

$$\begin{pmatrix} 0&5&0&0&1&0&0&1\\ 0&0&1&2&1&0&3&0\\ 3&0&0&0&0&0&1&3\\ 0&0&3&0&2&0&0&2\\ 1&1&0&2&0&2&0&1\\ 2&0&3&0&2&0&0&0\\ 1&0&0&3&0&3&0&0\\ 0&1&0&0&1&2&3&0 \end{pmatrix}$$

Fig. 1. Schedule whose coe-value is 140 and its coe-matrix

The *carry-over effects value* (coe-value for short) is defined as $\sum_{i,j}({c_{ij}}^2)$.

The *carry-over effects value minimization problem* is to find a schedule of which coe-value is minimum. Obviously, the coe-value of a schedule of n teams attains the lower bound $n(n-1)$ when all non-diagonal elements of the corresponding coe-matrix are 1. Such schedules are called *balanced*. In a balanced schedule, carry-over effects spread as evenly as possible. Fig. 2 shows a balanced schedule for $n = 8$ and its coe-value is $n(n-1) = 56$, while the coe-value of Fig. 1 is 140.

Russell [2] proposed the carry-over effects value minimization problem, and an algorithm for constructing a balanced schedule when n is a power of two. In addition, it is conjectured that there is no balanced schedule unless n is a power of two. This conjecture is still open; for $n = 6$ and 10, it was verified by computation.

Russell also proposed a constructive heuristic algorithm to obtain schedules of small coe-values for $n = p^m + 1$, where p is an odd prime and $m \geq 1$. (Note that when $n \leq 20$, every even n is either a power of two or the form $p^m + 1$.) The heuristic algorithm produces schedules whose coe-values are $60, 138, 196, 260, 428$ and 520 for $n = 6, 10, 12, 14, 18$ and 20, respectively. For $n = 6$, the schedule by Russell is indeed optimal. However, for $n = 10$ and 12, better schedules were recently obtained. For $n = 10$, Trick [3] reported a schedule whose coe-value is 122; for $n = 12$, Henz, Müller and Thiel [1] did a schedule whose coe-value is 188 (see Table 1). Both of them used constraint programming, and with constraint programming it seems difficult to find better schedules for larger n in practical computational time.

In the following, we propose a simple heuristic algorithm, which quickly generates better schedules than the previous best ones for $n \geq 14$. Our algorithm exploits the *circle method* (or polygon method), a well-known algorithm for constructing a round-robin tournament schedule. However, it has not yet been discussed in the context of minimizing the carry-over effects value.

The algorithm of the circle method is as follows:
in slot s ($s \in \{1, 2, \ldots, n-1\}$),
- team n plays team s;
- team i plays team j for $(i + j) \equiv 2s \bmod (n-1)$.

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

Fig. 2. Balanced schedule

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

Fig. 3. Schedule with the circle method

In the rest of this abstract, we denote the schedule created with the circle method by $\mathcal{C}_n$, where n is the number of teams. In $\mathcal{C}_n$, on the assumption that playing itself means playing team n, every team except n plays matches in the following order or its cyclic permutation: $1, 3, \ldots, n-1, 2, 4, \ldots, n-2$ (see Fig. 3). Accordingly, the coe-value of $\mathcal{C}_n$ is very large. (For instance, the coe-value of $\mathcal{C}_{10}$ is 468. We conjecture that $\mathcal{C}_n$ gives an optimal solution for *maximizing* the coe-value.)

Consider permuting of the columns, i.e. slots, of $\mathcal{C}_n$. For a permutation σ on the set of the slots $\{1, 2, \ldots, n-1\}$, we construct the schedule $\mathcal{C}_n(\sigma)$ whose s-th column is the $\sigma(s)$-th column of $\mathcal{C}_n$. In addition, we define the sequence $p(\sigma)$ as follows: $p(\sigma) = (\sigma(2) - \sigma(1), \sigma(3) - \sigma(2), \ldots, \sigma(n-1) - \sigma(n-2), \sigma(1) - \sigma(n-1)) \bmod (n-1)$. For a permutation σ, it is observed that if some elements of $p(\sigma)$ has a same value, particular elements of the coe-matrix of $\mathcal{C}_n(\sigma)$ increase; for instance, the schedule $\mathcal{C}_n$, which has a large coe-value, corresponds to the identical permutation and $p(\sigma) = (1, 1, \ldots, 1)$. Thus, we expect that a good schedule is obtained by a permutation σ such that elements of $p(\sigma)$ are as different as possible.

To obtain schedules of small coe-values, we generated a number of permutations σ randomly. Our algorithm produced schedules whose coe-values are 254, 412 and 496 for $n = 14$, 18 and 20 respectively, in less than 1 second (CPU: Pentium III 1.0 GHz, RAM: 1024 MB); all of which are better than the previous results. The best coe-values after two days of random generation are 400 and 488 for $n = 18$ and 20, respectively (see Table 1).

Finally, it should be noted that we obtained schedules whose coe-values are 108 and 176 for $n = 10$ and 12, respectively (Table 1). These results were achieved by adding constraints to the constraint programming formulation proposed by Trick [3]. Due to the space limitation, the detail is omitted.

References

1. Henz, M., Müller, T., Thiel, S.: Global constraints for round robin tournament scheduling. European Journal of Operational Research **153** (2004) 92–101

Table 1. Upper bounds of the carry-over effects value

#teams	old best	(status)	our results
4	12	([2], balanced)	
6	60	([2], optimal)	
8	56	([2], balanced)	
10	122	([3])	**108**
12	188	([1])	**176**
14	260	([2])	**254**
16	240	([2], balanced)	
18	428	([2])	**400**
20	520	([2])	**488**

2. Russell, K.G.: Balancing carry-over effects in round robin tournaments. Biometrika **67** (1980) 127–131
3. Trick, M.A.: A schedule-then-break approach to sports timetabling. In: Burke, E., Erben, W. (eds.): Practice and Theory of Automated Timetabling III (PATAT 2000, Konstanz, Germany, August, selected revised papers). Lecture Notes in Computer Science, Vol. 2079. Springer-Verlag, Berlin Heidelberg New York (2001) 242–253