The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 18, No. 4, pp.177-182, Aug. 31, 2018. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2018.18.4.177
JIIBC 2018-4-25

라운드-로빈 홈 앤드 어웨이 스포츠 리그 대진표 작성 정규형 라틴 방진 알고리즘

Canonical Latin Square Algorithm for Round-Robin Home-and-Away Sports Leagues Scheduling

이상운*

Sang-Un Lee*

요 약 최소 제동 수를 갖는 홈 앤드 어웨이 라운드-로빈 경기일정 대진표를 작성하는 문제는 매우 어려워 NP-난제로 알려져 있다. 본 논문에서는 임의의 팀 수 n에 대해서도 항상 동일한 패턴으로 경기일정 대진표를 $O(n)$ 수행 복잡도로 컴퓨터 프로그램 도움 없이 직접 손으로 작성할 수 있는 알고리즘을 제안하였다. 제안된 알고리즘은 $n=even$팀에 대해 $n\times n$ 정규형 라틴 방진을 작성하여 대진표를 작성하고, 최소 제동 수가 $n-2$가 되도록 홈-어웨이를 배정하였다. 또한, $n=odd$에 대해서는 $n=even$ 결과에서 최대 제동 수를 갖는 n번째 팀을 삭제하는 방법으로 제동이 전혀 없는 대진표를 작성하였다.

Abstract The home-and-way round-robin sports leagues scheduling problem with minimum brake is very hard to solve in polynomial time. This problem is NP-hard, the complexity status is not yet determined. This paper suggests round-robin sports leagues scheduling algorithm not computer-aided program but by hand with $O(n)$ time complexity for arbitrary number of teams n with always same pattern. The algorithm makes a list of mathes using $n\times n$ canonical latin square for $n=even$teams. Then trying to get home(H) and away(A) with $n-2$ minimum number of brakes. Also, we get the $n=odd$scheduling with none brakes delete a team own maximum number of brakes from $n=even$scheduling.

Key Words : sports leagues scheduling, round-robin, home-and-away, brake, canonical latin square

Ⅰ. 서 론

n개 팀으로 구성된 스포츠 경기(야구, 배구, 농구, 럭비, 축구 등)는 일반적으로 각 팀이 나머지 $n-1$개 팀과 돌아가면서 한 번씩 같은 시합수로 서로 대전하여 그 성적에 따라 순위를 결정하는 리그전(league match)을 치른다. 이를 라운드-로빈 방식이라 한다. 리그전의 장점은 참가 팀에게 평등하게 시합 할 수 있는 기회를 준다는 점이며, 단점은 결정된 대진표에 따라 예선전부터 연거푸 이기면 결승까지 올라가는 방식인 토너먼트(tournament) 방식에 비해 순위를 결정하기 까지 시간이 걸린다는 점이다. 내셔널 리그, 아메리칸 리그, 프로야구, 프로축구,

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 : 2018년 5월 20일, 수정완료 : 2018년 7월 1일
게재확정일자 : 2018년 8월 10일

Received: 20 May, 2018 / Revised: 1 July, 2018 /
Accepted: 10 August, 2018
*Corresponding Author: sulee@gwnu.ac.kr
Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

한국배구 슈퍼리그 등이 이 방식을 채택하고 있다.

n팀 리그전의 경기 횟수는 K_n-완전 그래프의 간선 수인 $\frac{n(n-1)}{2}$회이다. 예를 들면, $n=8$인 경우 $(8\times 7)/2=28$경기가 치러지며, 각 팀은 $n-1$팀과 경기를 치르므로 $d=n-1=7$일 동안, 각 일자별로 $m=n/2=4$ 경기(match)가 치러진다.

리그전은 한 곳에서 모든 경기를 수행하는 아마추어 경기와 지역연고팀이 홈앤드어웨이 방식의 프로경기가 있다. 이러한 경기에 대해 경기일정을 짜는 문제를 라운드-로빈 스포츠 경기 일정 문제(round-robin sports leagues scheduling problem, RSLSP)라 한다.[1,2] 또한, Ramussen과 Trick[2]에 따르면 홈-홈(H-H) 또는 어웨이-어웨이(A-A)가 연속하여 발생하는 경기흐름에 제동(brake)을 거는 경우를 최소화 시키는 구성법(constructive method), 최소 제동 제약을 가하는 문제, 여행거리 최소화 문제 등이 있음을 제시하였다.

RSLSP에서 팀 수 n은 odd(홀수)인 경우와 $even$(짝수)인 경우가 발생할 수 있다. 일반적으로 홈 경기장이 없는 아마추어 경기에 대해서는 홈앤드어웨이를 고려할 필요가 없어 팀 배정 일정을 짜는데 큰 어려움이 없다. 이는 Kirkman 순환(circle)법[3], 사전편찬 순서로 배정하는 탐욕 라운드-로빈 알고리즘, 정규형 라운드로빈 알고리즘[1]으로 해결할 수 있다.

그러나 홈 앤드 어웨이 방식 경기 대진표를 작성하는 문제에 대해서는 NP-난제(NP-hard)으로 다항시간 알고리즘이 알려져 있지 않다.[5] 따라서 다항시간 알고리즘이 존재하지 않다고 가정하고 메타휴리스틱 기법과 다항시간으로 풀 수 있는 탐욕기법들이 제안되고 있다. 메타휴리스틱 기법으로는 Rutjanisarakul과 Jiarasuksakun[6]의 유전자 알고리즘(genetic algorithm, GA), Hamiez와 Hao[7]의 타부 탐색법(Tabu search, TS)이 있으며, 다항시간 탐욕 알고리즘으로는 Werra[4]의 거리 패턴 간선 색칠법, Choi et al.[8]의 패턴법, Hamiez와 Hao[9]의 전수 수리법(exhaustive repair method), Miyashiro와 Matsui[5]의 2SAT(2-satisfiability problem)법, 그리고 최대 절단(max cut)법[10-13] 등이 있다.

본 논문에서는 최소 제동 수를 갖는 홈 앤드 어웨이 경기에 대한 $n=even$ 표준형$(n=8)$ 대진표를 $O(n)$의 선형시간으로 작성할 수 있는 방법을 제안한다. 이로부터 $n=odd(n-1)$의 대진표는 쉽게 유도할 수 있다. 따라서 표준형 대진표 작성에 대해 $n=even(4,6)$으로 감소시키거나 $n=even(10,12,\cdots)$로 증가시키면 어떠한 팀 수 n에 대해서도 적용할 수 있는 범용 알고리즘으로 활용할 수 있다. 2장에서는 최소 단절을 가진 홈-어웨이 대진표 연구결과를 고찰해 보고, 본 논문에서 제안될 라틴 방진 개념을 제시한다. 3장에서는 최소 제동 수를 갖는 라틴 방진법을 제안하여 라운드-로빈 스포츠 리그전 경기에 활용할 수 있는 방법을 제안한다.

II. 라운드-로빈 스포츠 경기 일정과 알고리즘 관련 분야

$n=8$인 경우의 홈 앤드 어웨이 라운드-로빈 스포츠 경기일정 대진표에 대한 연구 결과는 그림 1에 제시하였다. 여기서 { }는 집합기호, (i,j)는 호로, $i\to j$를 의미한다. 이는 T_j팀 홈 경기장에서 $\{T_i,T_j\}$ 경기를 하는 것으로 T_i는 어웨이(A), T_j는 홈(H)이 된다. 여기서 정준 1-인수분해(canonical 1-factorization)는 $d_i, i=1,2,\cdots,n-1$에 $\{n,i\}$와 $\{(i+k)\mathrm{mod}(n-1),(i-k)\mathrm{mod}(n-1)\}, k=1,2,\cdots,n/2-1$이 포함되도록 하는 방법이다.

(a) Canonical[1]	(b) Pattern[8]
d_1={(1,8),(2,7),(4,5),(6,3)}	d_1={(2,1),(3,8),(7,4),(5,6)}
d_2={(3,1),(5,6),(7,4),(8,2)}	d_2={(1,3),(4,2),(8,5),(6,7)}
d_3={(1,5),(3,8),(4,2),(6,7)}	d_3={(4,1),(2,6),(5,3),(7,8)}
d_4={(2,6),(5,3),(7,1),(8,4)}	d_4={(1,5),(8,2),(3,7),(6,4)}
d_5={(1,2),(3,7),(5,8),(6,4)}	d_5={(6,1),(2,3),(4,8),(7,5)}
d_6={(2,3),(4,1),(7,5),(8,6)}	d_6={(1,7),(5,2),(3,4),(8,6)}
d_7={(1,6),(3,4),(5,2),(7,8)}	d_7={(8,1),(2,7),(6,3),(4,5)}

그림 1. 라운드-로빈 스포츠 경기 일정
Fig. 1. Round-robin sports leagues scheduling

그림 1의 결과에 대해 홈(H)과 어웨이(A)를 표기하여 'H-H' 또는 'A-A'의 제동이 발생하는지 살펴본 결과는 그림 2와 같다. 제동 수는 당일 경기 수 $m=n/2$에 대해 $2m-2$회 발생한다고 알려져 있다. 따라서 $n=even$인 경우 $n-2$회, $n=odd$인 경우 $(n-1)-2$회 발생한다.

Canonical [1]		Team							
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
경기일	d_1	8(A)	7(A)	6(H)	5(A)	4(H)	3(A)	2(H)	1(H)
	d_2	3(H)	**8(H)**	**1(A)**	7(H)	6(A)	5(H)	4(A)	2(A)
	d_3	5(A)	**4(H)**	**8(A)**	2(A)	1(H)	7(A)	6(H)	3(H)
	d_4	7(H)	6(A)	5(H)	**8(H)**	**3(A)**	2(H)	1(A)	4(A)
	d_5	2(A)	1(H)	7(A)	**6(H)**	**8(A)**	4(A)	3(H)	5(H)
	d_6	4(H)	3(A)	2(H)	1(A)	7(H)	**8(H)**	**5(A)**	6(A)
	d_7	6(A)	5(H)	4(A)	3(H)	2(A)	**1(H)**	**8(A)**	7(H)
제동 수		**0**	**1**	**1**	**1**	**1**	**1**	**1**	**0**

Pattern [8]		Team							
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
경기일	d_1	2(H)	1(A)	8(A)	7(H)	6(A)	5(H)	4(A)	3(H)
	d_2	3(A)	4(H)	**1(H)**	**2(A)**	8(H)	7(A)	6(H)	5(A)
	d_3	4(H)	6(A)	**5(H)**	**1(A)**	3(A)	2(H)	8(A)	7(H)
	d_4	5(A)	8(H)	7(A)	6(H)	**1(H)**	**4(A)**	3(H)	2(A)
	d_5	6(H)	3(A)	2(H)	8(A)	**7(H)**	**1(A)**	5(A)	4(H)
	d_6	7(A)	5(H)	4(A)	3(H)	2(A)	8(H)	**1(H)**	**6(A)**
	d_7	8(H)	7(A)	6(H)	5(A)	4(H)	3(A)	**2(H)**	**1(A)**
제동 수		**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**

그림 2. 라운드-로빈 일정의 제동
Fig. 2. Brake of round-robin scheduling

The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 18, No. 4, pp.177-182, Aug. 31, 2018. pISSN 2289-0238, eISSN 2289-0246

여기서는 $n=8$이므로 제동 수는 6회가 발생한다. 두 방법 모두 제동 수는 6회로 대진표가 옳게 작성되었음을 알 수 있다.

공정한 팀 배정과 관련하여, A-A의 제동이 발생한 팀은 '불리'하다고 하며, 'H-H' 제동이 발생한 팀은 '유리'하다고 가정한다. 또한, 'A-A'와 'H-H'이 발생한 경우 또는 'A-A'나 'H-H'이 전혀 없는 팀은 '공정'하다고 가정한다. 이 가정에 따르면 그림 2의 정규형 방식[1] 결과는 T_1과 T_8은 '공정'으로, T_2, T_4, T_6은 '유리', T_3, T_5, T_7은 '불리'한 경우로 해석될 수 있다. 따라서 공정:유리:불리=2:3:3이다. 패턴방식[8] 결과는 T_1과 T_2는 '공정'으로, T_3, T_5, T_7은 '유리', T_4, T_6, T_8은 '불리'한 경우로 해석될 수 있다. 따라서 공정:유리:불리=2:3:3이다.

Miyashiro와 Matsui[5]는 $O(n^3)$ 수행 복잡도를 갖는 2SAT법으로 홈 앤드 어웨이를 결정하는 알고리즘을 제안하였다.

라틴 방진법은 $n \times n$ 행렬에 서로 다른 라틴어 문자 n개를 각 라틴어 문자는 행에서 정확히 한번만, 열에서 정확히 한 번만 존재하도록 채우는 문제이다.[14] 라틴어 문자를 상대 팀 번호 숫자로 변환시키고, 행을 경기 일, 열을 팀 번호로 결정한 $n \times n$ 행렬을 채우면 RSLSP는 라틴 방진법이 된다. 3장에서는 $n \times n$ 라틴 방진에 대해 상대 팀을 배정하고, 홈 앤드 어웨이를 결정하는 방법을 제안한다. 이 방법은 $n \times n$ 라틴 방진에 대해 상대팀 번호(숫자)를 채우고, H나 A를 결정하는 방법으로 $O(n)$ 수행 복잡도로 수행될 수 있다.

Ⅲ. 정규형 라틴 방진 알고리즘

본 장에서는 $n=even$과 odd 모두를 고려한다. 기본적으로 $n=8$과 7을 제시하며, 이로부터 n을 증가시키거나 감소시켜 원하는 팀의 수에 적합하도록 경기 일정을 설계할 수 있도록 정규형 라틴 방진법(canonical latin square, CLS)을 적용하였다. 먼저, $n=even$에 대한 팀 편성이 된 대진표는 $O(n)$ 수행 복잡도로 다음과 같이 작성되며, 그림 3에 제시되어 있다.

[$n=even$, $n-1 \times n-1$**라틴 방진**]

for $i=1$ to $n-1$
  $(i,j) \leftarrow i(i=j)$위치, $1,2,\cdots,i,i+1,\cdots,n-1$을 우에서 좌(역순)로 배정
end
for $i=1$ to $n-1$
  $(i,j) \leftarrow n(i=j)$
end
for $i=1$ to $n-1$
  $(i,n) \leftarrow i$
end

1행을 팀 번호로, 설정하면 나머지 $n-1 \times n$ 라틴 방진의 각 셀 값은 경기를 치를 상대 팀 번호가 되며, $n-1$행은 경기 일수가 된다. 이와 같이 배정된 결과는 두 팀이 항상 경기를 치를 짝으로 결정되어 각 경기 일에서는 $n/2$ 경기가 치러진다.

1	7	6	5	4	3	2
3	**2**	1	7	6	5	4
5	4	**3**	2	1	7	6
7	6	5	**4**	3	2	1
2	1	7	6	**5**	4	3
4	3	2	1	7	**6**	5
6	5	4	3	2	1	**7**

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

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

		Team							
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
Day	d_1	8	7	6	5	4	3	2	1
	d_2	3	8	1	7	6	5	4	2
	d_3	5	4	8	2	1	7	6	3
	d_4	7	6	5	8	3	2	1	4
	d_5	2	1	7	6	8	4	3	5
	d_6	4	3	2	1	7	8	5	6
	d_7	6	5	4	3	2	1	8	7

그림 3. $n=8$ 기본 형 대진표
Fig. 3. List of matches for $n=8$ basic type

다음으로 홈 앤드 어웨이 경기를 배정하는 방법은 $O(n)$ 수행 복잡도로 다음과 같이 수행되며, 그림 4에 제시되어 있다.

for $i=1$ to $n-1$
  $(i,j)=n(i=j)$좌측 셀에 A(또는 H), 우측 셀에 H(또는 A) 배정
  단, $i=n-1$인 경우 우측 셀인 $(n-1,n)$에는 배정하지 않는다.
end
for $i=2$ to $n-2$
  $(i,n) \leftarrow \overline{(i-1,n)}$/* H ↔A, A↔H : $(i-1,n)$과 반대인 H(또는 A)를 (i,n)에 배정*/
  $(i,j), i=j \leftarrow \overline{(i,n)}$/* H ↔A, A↔H */
end
for $j=1$ to $n-1$
  미배정된 우상삼각행렬과 좌하삼각행렬에 대해 기 배정된 셀의 H(A)와 반대인 A(H)를 반복하면서 배정.
end

1^{st} 배정	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
d_1	8	7(H)	6	5	4	3	2	1(A)
d_2	3(A)	8	1(H)	7	6	5	4	2
d_3	5	4(A)	8	2(H)	1	7	6	3
d_4	7	6	5(A)	8	3(H)	2	1	4
d_5	2	1	7	6(A)	8	4(H)	3	5
d_6	4	3	2	1	7(A)	8	5(H)	6
d_7	6	5	4	3	2	1(A)	8	7

2^{nd} 배정	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
d_1	8(H)	7(H)	6	5	4	3	2	1(A)
d_2	3(A)	8(A)	1(H)	7	6	5	4	2(H)
d_3	5	4(A)	8(H)	2(H)	1	7	6	3(A)
d_4	7	6	5(A)	8(A)	3(H)	2	1	4(H)
d_5	2	1	7	6(A)	8(H)	4(H)	3	5(A)
d_6	4	3	2	1	7(A)	8(A)	5(H)	6(H)
d_7	6	5	4	3	2	1(A)	8(H)	7(H)

3^{rd} 배정	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
d_1	8(H)	7(H)	6(A)	5(H)	4(A)	3(H)	2(A)	1(A)
d_2	3(A)	8(A)	1(H)	7(A)	6(H)	5(A)	4(H)	2(H)
d_3	5(H)	4(A)	8(H)	2(H)	1(A)	7(H)	6(A)	3(A)
d_4	7(A)	6(H)	5(A)	8(A)	3(H)	2(A)	1(H)	4(H)
d_5	2(H)	1(A)	7(H)	6(A)	8(H)	4(H)	3(A)	5(A)
d_6	4(A)	3(H)	2(A)	1(H)	7(A)	8(A)	5(H)	6(H)
d_7	6(H)	5(A)	4(H)	3(A)	2(H)	1(A)	8(H)	7(A)

결과	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
d_1	8(H)	7(H)	6(A)	5(H)	4(A)	3(H)	2(A)	1(A)
d_2	3(A)	8(A)	1(H)	7(A)	6(H)	5(A)	4(H)	2(H)
d_3	5(H)	4(A)	8(H)	2(H)	1(A)	7(H)	6(A)	3(A)
d_4	7(A)	6(H)	5(A)	8(A)	3(H)	2(A)	1(H)	4(H)
d_5	2(H)	1(A)	7(H)	6(A)	8(H)	4(H)	3(A)	5(A)
d_6	4(A)	3(H)	2(A)	1(H)	7(A)	8(A)	5(H)	6(H)
d_7	6(H)	5(A)	4(H)	3(A)	2(H)	1(A)	8(H)	7(A)

=

결과	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
d_1	8(A)	7(A)	6(H)	5(A)	4(H)	3(A)	2(H)	1(H)
d_2	3(H)	8(H)	1(A)	7(H)	6(A)	5(H)	4(A)	2(A)
d_3	5(A)	4(H)	8(A)	2(A)	1(H)	7(A)	6(H)	3(H)
d_4	7(H)	6(A)	5(H)	8(H)	3(A)	2(H)	1(A)	4(A)
d_5	2(A)	1(H)	7(A)	6(H)	8(A)	4(A)	3(H)	5(H)
d_6	4(H)	3(A)	2(H)	1(A)	7(H)	8(H)	5(A)	6(A)
d_7	6(A)	5(H)	4(A)	3(H)	2(A)	1(H)	8(A)	7(H)

그림 4. $n=8$ 기본 형 홈 앤드 어웨이 결정
Fig. 4. Decision of home and away for $n=8$ basic type

배정 결과를 보면, $n=even$에 대해 각 팀은 $n-1$ 팀과 경기를 수행하므로 H-A를 반복하여 배정하면 n번째 팀이 중간에 위치한 $n-2$개 팀은 'H-()-A'나 'A-()-H' 형태가 되어 제동이 발생한다. 따라서 $n=even$인 경우 최소 제동 수는 $n-2$개다.

$n=odd$인 경우 $n=even$에서 제동이 최대로 발생하는 팀(여기서는 $T_j, j=n$)을 삭제하고, 나머지 팀이 n번째 팀과 치를 경기를 취소시키면 다음과 같이 쉽게 해결되며, 그림 5에 제시되어 있다. 이 과정 또한 $O(n)$ 수행 복잡도로 수행된다. 여기서는 그림 2의 정규형[1]과 패턴형[8]에 대해서도 변환 결과를 제시하였으며, 3가지 모델 모두 제동이 전혀 없는 결과를 얻었다.

[$n=odd$]

$n=even$에서 제동이 최대로 발생하는 팀(여기서는 $T_j, j=n$) 삭제

나머지 팀이 n번째 팀과 치를 경기를 취소

Day	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
	Team							
d_1	-	7(H)	6(A)	5(H)	4(A)	3(H)	2(A)	-
d_2	3(A)	-	1(H)	7(A)	6(H)	5(A)	4(H)	-
d_3	5(H)	4(A)	-	2(H)	1(A)	7(H)	6(A)	-
d_4	7(A)	6(H)	5(A)	-	3(H)	2(A)	1(H)	-
d_5	2(H)	1(A)	7(H)	6(A)	-	4(H)	3(A)	-
d_6	4(A)	3(H)	2(A)	1(H)	7(A)	-	5(H)	-
d_7	6(H)	5(A)	4(H)	3(A)	2(H)	1(A)	-	-

Day	T_1	T_2	T_3	T_4	T_5	T_6	T_7
	Team						
d_1	-	7(H)	6(A)	5(H)	4(A)	3(H)	2(A)
d_2	3(A)	-	1(H)	7(A)	6(H)	5(A)	4(H)
d_3	5(H)	4(A)	-	2(H)	1(A)	7(H)	6(A)
d_4	7(A)	6(H)	5(A)	-	3(H)	2(A)	1(H)
d_5	2(H)	1(A)	7(H)	6(A)	-	4(H)	3(A)
d_6	4(A)	3(H)	2(A)	1(H)	7(A)	-	5(H)
d_7	6(H)	5(A)	4(H)	3(A)	2(H)	1(A)	-

(a) Canonical latin square method

Canonical [1]		Team						
		T_1	T_2	T_3	T_4	T_5	T_6	T_7
경기일	d_1	-	7(A)	6(H)	5(A)	4(H)	3(A)	2(H)
	d_2	3(H)	-	1(A)	7(H)	6(A)	5(H)	4(A)
	d_3	5(A)	4(H)	-	2(A)	1(H)	7(A)	6(H)
	d_4	7(H)	6(A)	5(H)	-	3(A)	2(H)	1(A)
	d_5	2(A)	1(H)	7(A)	6(H)	-	4(A)	3(H)
	d_6	4(H)	3(A)	2(H)	1(A)	7(H)	-	5(A)
	d_7	6(A)	5(H)	4(A)	3(H)	2(A)	1(H)	-
제동 수		0	1	1	1	1	1	1

(b) Canonical method

Pattern [8]		Team						
		T_2	T_3	T_4	T_5	T_6	T_7	T_8
경기일	d_1	-	8(A)	7(H)	6(A)	5(H)	4(A)	3(H)
	d_2	4(H)	-	2(A)	8(H)	7(A)	6(H)	5(A)
	d_3	6(A)	5(H)	-	3(A)	2(H)	8(A)	7(H)
	d_4	8(H)	7(A)	6(H)	-	4(A)	3(H)	2(A)
	d_5	3(A)	2(H)	8(A)	7(H)	-	5(A)	4(H)
	d_6	5(H)	4(A)	3(H)	2(A)	8(H)	-	6(A)
	d_7	7(A)	6(H)	5(A)	4(H)	3(A)	2(H)	-
제동 수		0	1	1	1	1	1	1

(c) Pattern method

그림 5. $n=odd$ 대진표
Fig. 5. List of mathes for $n=odd$

정규형 라틴 방진법으로 얻은 경기 대진표 결과를 살펴보면, $n=even$인 경우, T_1과 T_8은 '공정'으로, T_2, T_4, T_6는 '불리', T_3, T_5, T_7은 '유리'한 경우이다. 따라서 공정:유리:불리=2:3:3에 대해, 그림 6과 같이 전년도 성적에 따라 팀을 배정하면 4,1,8,2,7,3,6,5 성적순으로 팀을 배정할 수 있다. $n=odd$인 경우 제동이 전혀 발생하지 않아 팀은 랜덤으로 배정해도 무관함을 알 수 있다.

$n=even$		Team							
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
Day	d_1	8(H)	7(H)	6(A)	5(H)	4(A)	3(H)	2(A)	1(A)
	d_2	3(A)	8(A)	1(H)	7(A)	6(H)	5(A)	4(H)	2(H)
	d_3	5(H)	4(A)	8(H)	2(H)	1(A)	7(H)	6(A)	3(A)
	d_4	7(A)	6(H)	5(A)	8(A)	3(H)	2(A)	1(H)	4(H)
	d_5	2(H)	1(A)	7(H)	6(A)	8(H)	4(H)	3(A)	5(A)
	d_6	4(A)	3(H)	2(A)	1(H)	7(A)	8(A)	5(H)	6(H)
	d_7	6(H)	5(A)	4(H)	3(A)	2(H)	1(A)	8(H)	7(A)
제동 수		0	1	1	1	1	1	1	0
유·불리 유무		공정	불리	유리	불리	유리	불리	유리	공정
전년도 성적에 따른 팀 배정		4	1	8	2	7	3	6	5

$n=odd$	Team						
	T_1	T_2	T_3	T_4	T_5	T_6	T_7
d_1	-	7(H)	6(A)	5(H)	4(A)	3(H)	2(A)
d_2	3(A)	-	1(H)	7(A)	6(H)	5(A)	4(H)
d_3	5(H)	4(A)	-	2(H)	1(A)	7(H)	6(A)
d_4	7(A)	6(H)	5(A)	-	3(H)	2(A)	1(H)
d_5	2(H)	1(A)	7(H)	6(A)	-	4(H)	3(A)
d_6	4(A)	3(H)	2(A)	1(H)	7(A)	-	5(H)
d_7	6(H)	5(A)	4(H)	3(A)	2(H)	1(A)	-
제동 수	0	0	0	0	0	0	0
유·불리 유무	공정	공정	공정	공정	공정	공정	공정
팀 배정	랜덤 배정						

그림 6. 팀 배정
Fig. 6. Assignment of teams

Ⅳ. 알고리즘 적용성 평가

n팀에 대한 경기 대진표가 알려져 있지 않은 경우의 라운드 로빈 경기 대진표를 작성하고자 할 때 가장 쉬우면서도 일정한 규칙에 따라 대진표를 작성해야만 한다. 그러나 $n=even$에 대해 정규형[1]이나 패턴형[8]을 찾는 과정은 쉽지 않아 알고리즘 전문가가 아니면 쉽게 적용할 수 없는 점이 문제로 대두된다. 반면에, 본 논문에서 제안된 라틴 방진 법은 n이 어떠한 값을 갖더라도 쉽게 정해진 규칙에 따라 $n \times n$ 라틴 방진의 각 셀 숫자(경기 상대 팀)를 결정할 수 있으며, 홈 또는 어웨이 또한 정해진 규칙대로 쉽게 결정할 수 있는 장점을 갖고 있다.

$n=4,6,10,12$에 대해 제안된 라틴 방진법을 적용하여 $n=3,5,9,11$의 경기 대진표 까지 작성한 결과는 그림 7과 같다.

그림에서 알 수 있듯이 $n=even$이나 이로부터 유도된 $n=odd$ 모두 동일한 패턴을 갖고 있음을 알 수 있다. 따라서 n이 어떠한 값을 가지더라도 컴퓨터 프로그램의 도움 없이도 본 논문에서 제안된 정규형 라틴 방진을 만드는 방법을 적용하면 손으로 직접 경기일정 대진표를 작성할 수 있음을 알 수 있다. 또한 $n=even$인 경우 제동수는 $n-2$개를, $n=odd$인 경우 제동이 전혀 없는 대진표를 설계할 수 있다.

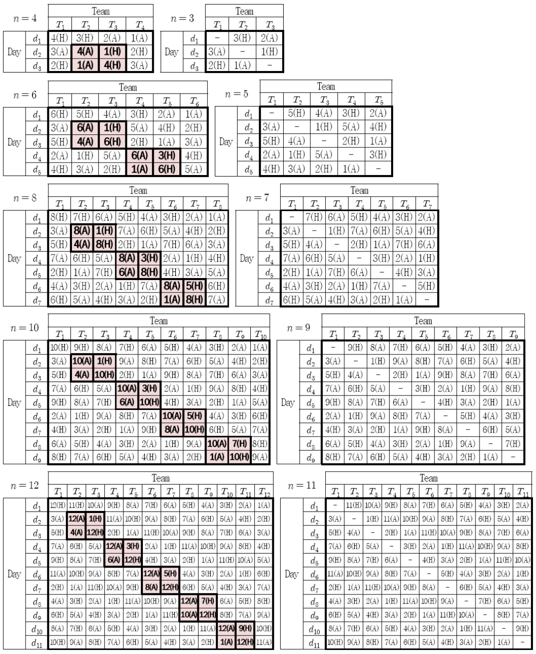

그림 7. 정규형 라틴 방진 경기일정 대진표
Fig. 7. Matching scheduling of canonical latin square

V. 결 론

본 논문에서는 홈 앤드 어웨이 라운드-로빈 방식의 경기를 수행하는 대부분의 프로 스포츠 경기 일정표를 작성하는 문제를 다루었다. 이 문제에 대해서는 홈-홈(H-H) 또는 어웨이-어웨이(A-A)가 연속해서 발생하는 제동이 가능한 없도록 해야만 한다. 왜냐하면 H-H 경기가 연속해서 발생하는 팀은 유리한 경기를 치를 수 있으며, A-A가 연속해서 발생하는 팀은 불리한 경기를 치를 수 있기 때문이다. 지금까지는 팀 수 $n=even$인 경우 최소 제동 수는 $n-2$로 알려져 있으며, $n=odd$인 경우의 제동 수는 알려져 있지 않다. 또한, $n=even$인 경우 최소 제동 수 $n-2$를 충족하는 경기 대진표 작성에 성공한 사례도 극히 드물다.

본 논문은 n이 어떠한 값을 가지더라도 제동 수 $n-2$를 갖는 $n=even$홈 앤드 어웨이 라운드-로빈 방식 경기 일정 대진표를 컴퓨터 프로그램의 도움 없이 직접 손으로 $O(n^2)$ 복잡도로 작성하는 방법인 정규형 라틴 방진법을 제안하였으며, 이로부터 제동 수가 전혀 발생하지 않는 $n=odd$로 단순히 변환시킬 수 있었다.

제안된 알고리즘은 자체 경기장을 갖고 있는 프로 팀의 홈 앤드 어웨이 방식에 채택할 수 있으며, 더 나아가 아마추어 팀과 같이 홈 앤드 어웨이 방식을 채택할 수 없는 경우에도 적용이 가능한 장점을 갖도록 설계되었다. 이를 위해서는 제안된 알고리즘의 팀 배정(Step 1 ~ Step 6) 단계만 수행하면 된다.

References

[1] R. M. R. Lewis, “A Guide to Graph Colouring Algorithms and Applications: Chapter 7. Designing Sports Leagues,” Springer, pp. 169-193, Oct. 2015, ISBN 978-3-319-25728-0, doi:10.1007/978-3-319-25730-3

[2] R. V. Rasmussen and M. A. Trick, “Round Robin Scheduling – A Survey,” European Journal of Operational Research, Vol. 188, No. 3, pp. 617-636, Aug. 2008, doi:10.1016/j.ejor.2007.05.046

[3] T. Kirkman, “On a Problem in Combinations,” Cambridge Dublin Math Journal, Vol. 2, pp. 191–204, 1847.

[4] D. de Werra, “Some Models of Graphs for Scheduling Sports Competitions,” Discrete Applied Mathematics, Vol. 21, No. 1, pp. 47-65, Sep. 1988, doi:10.1016/0166-218X(88)90033-9

[5] R. Miyashiro and T. Matsui, “A Polynomial-time Algorithm to Find an Equitable Home-Away Assignment,” Operational Research Letters, Vol. 33, No. 3, pp. 235-241, May 2005, doi:10.1016/j.orl.2004.06.004

[6] T. Rutjanisarakul and T. Jiarasuksakun, “A Sport Tournament Scheduling by Genetic Algorithm with Swing Method,” Cornell University Library, pp. 1-7, arXiv:1704.04879, Apr. 2017.

[7] J. P. Hamiez and J. K. Hao, “Solving the Sports League Scheduling Problem with Tabu Search,” Workshop on Local Search for Planning and Scheduling, pp. 24-36, Aug. 2000, doi:10.1007/3-540-45612-0_2

[8] S. B. Choi, S. S. Jeung, and T. Y. Han, “Home-Away Sports League Scheduling with Minimum Breaks,” Journal of Korean Society of Sports

Science, Vol. 24, No. 4, pp. 691-701, Aug. 2015, uci:G704- 001369.2015.24.4.098

[9] J. P. Hamiez and J. K. Hao, "A Linear-time Algorithm to Solve the Sports League Scheduling Problem(prob026 of CSPLib)," Discrete Applied Mathematics, Vol. 143, No. 1-3, pp. 252-265, Sep. 2004, doi:10.1016/j.dam.2003.10.009

[10] M. Elf, M. Junger, and G. Rinaldi, "Minimizing Breaks by Maximizing Cuts," Operational Research Letters, Vol. 31, No. 5, pp. 343-349, Sep. 2003, doi:10.1016/S0167-6377(03)00025-7

[11] M. X. Goemans and D. P. Williamson, "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems using Semidefinite Programming," Journal of ACM, Vol. 42, No. 6, pp. 1115-1145, Nov. 1995, doi:10.1145/227683.227684

[12] R. Miyashiro and T. Matsui, "Semidefinite Programming based Approaches to the Break Minimization Problem," Computers & Operations Research, Vol. 33, No. 7, pp. 1975-1982, Jul. 2006, doi:10.1016/j.cor.2004.09.030

[13] M. A. Trick, "A Schedule-then-Break Approach to Sports Timetabling," International Conference on the Practice and Theory of Automated Timetabling III, pp. 242-253, 2000.

[14] W. D. Wallis and J. C. George, "Introduction to Combinatorics," CRC Press, p. 212, 2011, ISBN 978-1 -4398-0623-4

저자 소개

이 상 운(정회원)

- 1987년 : 한국항공대학교 항공전자공학과 (학사)
- 1997년 : 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007.2 : 국립 원주대학 여성교양과 조교수
- 2007.3 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 최적화 알고리즘
- E-Mail : sulee@gwnu.ac.kr