

# 시뮬레이티드 어닐링 기법을 이용한 한국 프로야구 리그 일정계획 Scheduling for Korean Professional Baseball League Using Simulated Annealing : A case study

전지원, 이진호, 이영훈

연세대학교 정보산업공학과  
서울특별시 서대문구 신촌동 134 연세대학교 공과대학 정보산업공학과  
Fax : 02-364-7807, E-mail : youngh@yonsei.ac.kr

## Abstract

한국 프로야구 일정계획은 SLSP(Sports League Scheduling Problem)분야의 문제로서 전형적인 조합최적화 문제이다. 본 연구는 한국 프로야구 일정계획 문제의 제약 조건을 분석하고, 다양한 목적함수에 대해 검토하였다. 또한 이를 효과적으로 풀기 위하여 TTSA(Simulated Annealing for the Traveling Tournament)을 사용하였다. TTSA는 TTP(Traveling Tournament Problem) 문제를 풀기 위해 제안된 방법으로 가능해와 비가능해 영역을 동시에 탐색하여 빠른 시간 안에 좋은 해를 구할 수 있다. 실험은 2005년도 한국 프로야구 리그 일정을 바탕으로 하였으며, 그 결과 팀들의 이동 거리의 총합 최소화, 가장 이동거리가 긴 팀의 이동거리 최소화, 팀 간의 이동거리 차이 최소화 측면에서 각각 14%, 22%, 21% 더 좋은 일정을 얻을 수 있었다.

## 1. 서론

한국 프로야구 일정계획은 SLSP(Sports League Scheduling Problem) 분야에 속하는 문제로서 전형적인 조합최적화 문제이다. SLSP는 McAloon et al.(1997)에 의하여 제안된 이후 Operation Resarch영역에서 주요하게 다뤄지고 있다.

SLSP는 까다로운 제약조건들로 인해 그간 제약조건을 만족시키는 해를 찾기 위한 노력들이 주를 이루었다. 최근에는 이러한 제약조건들을 포함하면서 팀들의 이동거리 합을 최소화 라는 목적함수를 가지는 형태의 문제인 TTP(Traveling Tournament Problem)가 Easton et al.(2001)의하여 제안되었다.

TTP문제는 다음과 같이 정리할 수 있다.

- ▶ 입력 : n, 팀의 수; D, 팀 간의 거리 행렬;
- ▶ 결과 : n 팀간의 Double-Round Robin 방식의 경기 일정. 경기 일정은 다음의 조건을 만족시킨다.
  - 일정상의 모든 제약조건을 만족시킨다.
  - 각 팀의 이동거리의 합을 최소화한다.

Double-Round Robin 방식이란 각 팀이 다른 팀과 2번의 경기를 가지되, 한 경기는 홈에서 다른 한 경기는 원정에서 가지는 경기 방식을 말한다. Double-Round Robin 방식으로 경기를 치르게 되면 팀의 수가 n인 경우 2n-2 Round를 가지게 된다.

TTP 문제의 최적해를 구하기 위한 다양한 시도들이 연구되어 졌다. IP(Integer Programming)을 이용한 방법(Easton et al., 2001), Lagrangian Relaxtion을 이용한 방법(Benoist et al., 2001), Constraint Programming과 Integer Programming을 함께 이용하는 방법(Easton et al., 2003), 그리고 메타 휴리스틱을 이용한 방법(Anagnostopoulos et al., 2003)들이 연구되었다.

CP와 IP를 이용하여 Branch and Price 기법을 이용하여 해를 구하는 방법(Easton et al., 2003)은 해의 최적성을 보장해 주지만 상당한 계산비용이 소요된다.

메타 휴리스틱 기법을 이용한 방법은 짧은 계산시간 속에서 매우 좋은 해를 구할 수가 있으며, 작은 문제 사이즈의 경우에는 최적해에 해당하는 값을 구할 수가 있다.

특히 Simulated Annealing(SA)을 응용한 TTSA(Simulated Annealing for the Traveling Tournament Problem)(Anagnostopoulos et al., 2003)는 계산 시간과 해의 우수성 측면에서 매우 뛰어난 결과를 보여주고 있으며, 본 연구에서 이를 응용하였다. 이에 대한 자세한 내용은 뒤에서 살펴볼 것이다.

본 연구에서는 한국 프로야구 리그를 TTP의 확장 형태로 변형하고, 이를 TTSA 방법을 응용하여 2005년 한국 프로야구 리그의 일정 계획을 수립하고 기존의 일정계획과 비교해 보고자 한다.

## 2. 문제정의

한국 프로야구 리그는 1981년도에 6팀으로 시작하여, 현재 8팀으로 리그가 운영 중이다. 2005년도에는 4월2일(토) 개막을 시작으로 8월30일(화)까지 150일간의 열전에 돌입하며 팀당 126경기, 팀간 18차전, 총 504경기를 치른다.

한국 프로야구 리그의 일정계획을 수립하는데 필요한 주요 제약조건을 살펴보면 다음과 같다.

1. 각 팀의 126경기는 홈, 원정 모두 63경기씩이며, 팀간 18차전도 홈과 원정을 9경기씩 나누어 치른다.
2. 개막경기는 2004년도 한국야구선수권대회 1,2,3,4위팀의 구장에서 열린다.
3. 현재 2개의 팀이 같은 구장을 홈으로 사용하기 때문에 2팀이 동시에 홈경기를 가질 수 없다.
4. 홈 또는 원정의 연속 경기는 6연전 까지만 허용한다.
5. 3만명 이상의 관중을 수용할 수 있는 구장을 홈으로 가진 4개의 구단은 공휴일에 한번 이상 7개 구단과 홈 경기를 가진다.

6. 제주도에서 1경기를 가진다.
7. 월요일을 휴식일로 가진다. 단 월요일이 공휴일인 경우는 월요일 경기를 가지며, 휴식일을 조정된다.
8. 한국 프로야구 리그는 상대팀과의 경기는 3번의 연속 경기를 가진다.

각 팀 간 18차전은 한번 경기에 3연전을 하게 된다. 3연전을 1Round로 하면 팀들은 6번의 Round를 3번은 홈에서 3번은 원정에서 하게 된다. 이것은 홈에서 1번 원정에서 1번 경기를 갖는 Double-Round Robin방식의 경기를 3번 가지는 것과 같다. 따라서 한국 프로야구 리그는 TTP의 확장 형태가 되는 것이다.

제주도 경기의 제약조건의 경우는 제주도 경기를 선정하는 것이 별도의 문제로 생각해 볼 수 있기 때문에 이 연구에서는 고려하지 않는다.

따라서 본 문제에서는 나머지 제약 조건을 고려를 하되 5번 제약조건, 큰 구장을 소유한 구단의 휴일 경기에 대한 제약조건은 그 복잡성으로 인해 제약조건에서 제외를 하고, 1, 2, 3, 4, 7을 만족시키도록 하였다.

휴식일의 경우에는 팀의 이동경로를 홈을 거쳐서 이동하는 것으로 계산하였다. 즉, '원정-원정' 경기의 경우 휴식일이 끼어있는 경우에는 '원정-홈-원정'으로 팀의 이동경로를 조정하였다.

그리고 한국 프로야구 규정에는 명시되어 있지 않았지만, 되풀이경기 금지 제약조건을 본 문제에서는 추가 시켰다. 되풀이 경기란 특정 Round에서 "1-2"경기를 한 후에 바로 그 다음 라운드에 "2-1" 경기를 할 수 없다는 것이다. 즉 "1-2"번 팀이 1Round에 경기를 가졌다면 2Round에서는 "2-1"경기를 할 수 없는 것이다.

한국 프로야구의 일정은 <Table 1>와 같이 만들어진다. 각 팀은 각 Round에 3번의 연속경기를 가지게 되며, 총 42Round, 126경기를 가지게 된다. 표에서 음수 값은 원정 경기를 의미한다. <Table 1>의 경우 1팀은 1번째 Round에 2팀과 원정경기를 가지게 되고, 그 다음 7번 팀과 홈경기를 가지게 된다. 그리고 8팀과의 원정경기 3팀과의 홈경기가 된다. 이 경우 1팀의 이동 지역 순서는 "2-1-8-1...5-1-7-1"이 된다.

Table 1. 한국 프로야구 리그 일정표

	R1	R2	R3	R4	...	R39	R40	R41	R42
1	-2	7	-8	3	...	-5	4	-7	8
2	1	-5	6	-4	...	7	-6	5	-3
3	-4	6	5	-1	...	4	-8	-6	2
4	3	-8	-7	2	...	-3	-1	8	-5
5	-6	2	-3	8	...	1	7	-2	4
6	5	-3	-2	7	...	-8	2	3	-7
7	8	-1	4	-6	...	-2	-5	1	6
8	-7	4	1	-5	...	6	3	-4	-1

본 연구에서 고려하는 목적함수는 다음 3가지이다.

1. 각 팀 이동거리의 합의 최소화  
 각 팀들이 이동하는 거리의 합에 대한 최소값을 구하여 전체 리그에서 좋은 경기 일정을 구하기 위한 것이 목적이다.

2. 각 팀 이동거리의 차의 최소화  
 한 시즌 동안 각 팀이 이동하는 거리가 달라짐에 따라 발생하는 문제점을 해결하기 위해 이 목적함수를 이용하였다.

3. 팀 이동거리 중 최장 이동거리가 최소화  
 이 역시 한 시즌동안 각 팀이 이동하는 거리가 달라짐에

따라 발생하는 문제점을 해결하기 위한 것이다. 두 번째 목적함수와는 달리 최장 이동거리를 최소화함으로써 각 팀들의 이동거리에 균형을 맞추고자 한 것이다.

### 3. TTSA(Simulated Annealing for the Traveling Tournament Problem)

TTP문제를 풀기 위해 제안된 메타휴리스틱 방법으로 Simulated Annealing(SA)을 응용하는 방법 (Anagnostopoulos et al, 2003)이다. 이 방법은 짧은 시간 안에 매우 좋은 해를 구할 수 있으며 8개팀의 경우에는 일정 시간 안에 최적 값을 구할 수 있다.

이 방법의 특징은 제약 조건은 강한 제약조건(hard constraints)와 약한 제약조건(soft constraints)로 나누어서 강한 제약조건은 항상 만족시키도록 하며, 약한 제약조건 경우에는 가능해(feasible)와 불가능해(infeasible)를 모두 탐색한다는 것이다.

강한 제약 조건에 속하는 것은 Double-Round Robin의 경기 방식에 대한 것이다. 이것은 특정라운드에서 각 팀은 서로 다른 팀과 모두 경기를 가지는 것과 각 팀은 서로 다른 팀과 원정과 홈에서 각각 한 번씩 경기를 가지는 것이다.

약한 제약조건에 속하는 것은 연속경기, 즉 홈 혹은 원정 경기를 2혹은 3게임 이상 연속으로 치를 수 없다는 것과 같이 Double-Round Robin의 경기 방식을 제외한 리그 경기 운영 규칙이 된다.

TTSA에서 사용하는 SA의 변형된 알고리즘은 <Figure 1>와 같으며, 이것의 특징을 요약하면 다음과 같다.

1. 가능해와 불가능해를 모두 탐색한다.
2. 탐색한 해 중에 가장 좋은 가능해와 불가능해를 기억한다.
3. 새롭게 가장 좋은 가능해와 불가능해 나타나면 그 보상으로 모든 연산횟수 계산자를 초기화한다.
4. 지역 최적해 값을 벗어나기 위한 방법으로 재가열 방법을 이용한다.
5. 불가능해의 경우에는 일정한 가중치에 해당하는 벌점을 준다.
6. 해의 이동시에 새로 이동한 해가 불가능해/가능해인 경우에 따라 벌점 가중치를 증가/감소시킨다.

```

1. find random schedule S;
2. bestFeasible ← ∞; nbf ← ∞;
3. bestInfeasible ← ∞; nbi ← ∞;
4. reheat ← 0; counter ← 0; phase ← 0;
5. while reheat ≤ maxReheats do
6.   phase ← 0;
7.   while phase ≤ maxPhase do
8.     counter ← 0;
9.     while counter ≤ counterLimit do
10.      select a random move m from neighborhood(S);
11.      let S' be the schedule obtained from S with m;
12.      if C(S') < C(S) or
13.         nbv(S') == 0 and C(S') < bestFeasible or
14.         nbv(S') > 0 and C(S') < bestInfeasible
15.      then
16.        accept ← true;
17.      else
18.        accept ← true with probability exp(-ΔC/T),
19.           false otherwise;
20.      if accept then
21.        S ← S';
22.      if nbv(S') == 0 then
23.        nbf ← min(C(S), bestFeasible);
24.      else
25.        nbi ← min(C(S), bestInfeasible);
26.      if nbf < bestFeasible or nbi < bestInfeasible then
27.        reheat ← 0; counter ← 0; phase ← 0;
28.        bestTemperature ← T;
29.        bestFeasible ← nbf;
30.        bestInfeasible ← nbi;
31.        if nbv(S) == 0 then
32.          ω ← ω/θ;
33.        else
34.          ω ← ω*δ;
35.      counter++;
36.      phase++;
37.      T ← T*β;
38.      reheat++;
39.      T ← 2*bestTemperature;

```

Figure 1 . TTSA Algorithm

TTSA에서의 비용계산 방식은 다음과 같이 정리 할 수 있다.

$$C(S) = \begin{cases} cost(s) & \text{if } S \text{ is feasible} \\ \sqrt{cost(s)^2 + [w \cdot f(nbv(S))]^2} & \text{otherwise} \end{cases}$$

여기서  $nbv(S)$ 는 불가능해 S의 제약조건 위반사항의 개수이며,  $w$ 는 가중치 그리고  $f(nbv(S))$ 는  $f(1)=1$ 로 되는 선형적인 벌점 함수 이다.

본 연구에서는  $f(v) = 1 + \sqrt{v}nv/2$  으로 설정하여 사용하였다.

$w$ 는 불가능해에 대한 가중치로서 이  $w$ 값은 새로운 해로 이동시에 이 새로운 해가 가능해일 경우에는  $\theta$ 값으로 나누어 주며( $w/\theta$ ,  $\theta > 1$ ), 불가능해일 경우에는  $\delta$ 값으로 곱해 주어( $w*\delta$ ,  $\delta > 1$ ) 해가 불가능해에서 가능해로 이동할 수 있도록 하였다.

TTSA에서 이웃해의 생성은 다음과 같은 다섯 가지 방법을 통하여 생성하였다.

1. Swap Homes

두 팀을 골라 양 팀의 홈경기와 원정경기의 라운드를 서로 바꾸는 것이다. 예를 들어 4-2번팀의 Swap Home를 하게 되면 2번팀의 홈경기는 본래 5라운드이나 원정경기 일정인 8라운드와 서로 바꾸어 2-4번팀의 홈경기는 8라운드가 되고 5라운드가 원정경기가 되는 것이다. 4번 팀 역시 마찬가지로 2번팀과의 홈, 원정경기의 Round가 바뀌어진다.

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	4	3	6	-4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	-2	1	5	2	-6	-3
5	-2	-3	6	4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	4	3	6	-4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	2	1	5	-2	-6	-3
5	-2	-3	6	4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

Figure 2. 2-4팀의 Swap Homes

2. Swap Rounds

두개의 Round를 선택하여 각 라운드의 경기 모두를 바꾼다. 예를 들어 3, 5 라운드를 선택하였다 하면 3번 라운드에 해당하는 열의 내용과 5번 라운드에 해당하는 열의 내용을 모두 바꾼다.

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	4	3	6	-4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	-2	1	5	2	-6	-3
5	-2	-3	6	4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	-5	3	4	-4	-3	5	2	-6
2	5	1	4	-6	-3	3	6	-4	-1	-5
3	-4	5	6	-1	2	-2	1	-6	-5	4
4	3	6	-2	-5	-1	1	5	2	-6	-3
5	-2	-3	1	4	6	-6	-4	-1	3	2
6	-1	-4	-3	2	-5	5	-2	3	4	1

Figure 3. 3-5Round의 Swap Rounds

3. Swap Teams

2 팀을 선택하여 2팀의 경기 일정을 바꾸어준다. 예를 들어 2-5팀을 선택하게 되면, 2번팀의 경기 일정인 "5,1,-3,-6,4,3,5,-4,-1,-5"를 5번팀의 경기 일정으로 설정하고 마찬가지로 5번팀의 경기 일정을 2번팀의 경기 일정으로 설정한다. 이 때 2-5 서로간의 경기는 제외한다.

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	4	3	6	-4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	-2	1	5	2	-6	-3
5	-2	-3	6	4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-5	4	3	-2	-4	-3	2	5	-6
2	5	-3	6	4	1	-6	-4	-1	3	-5
3	-4	2	5	-1	6	-5	1	-6	-2	4
4	3	6	-1	-2	-5	1	2	5	-6	-3
5	-2	1	-3	-6	4	3	6	-4	-1	2
6	-1	-4	-2	5	-3	2	-5	3	4	1

Figure 4. 2-5팀의 Swap Teams

4. Partial Swap Rounds

두개의 라운드를 선택하고, 특정 팀을 선택하여 특정팀과 두 라운드에 해당하는 경기순서를 바꾸는 것이다. 예를 들어 2팀의 2, 9 Round를 선택하게 되면 "2-1"경기는

2Round에서 9Round로 “6-2”경기는 9Round에서 2Round로 바뀌는 것이다.

이 때 Round 열의 제약조건-각 팀은 서로다른팀과 경기를 해야한다-을 만족시키기 위하여, 다른 팀의 경기 일정도 바뀌게 된다. 예를 들어 2팀은 2Round에 6팀과 경기를 하게 되었으므로 본래 2Round에 6팀과 경기를 하는 4팀은 6팀과 경기를 할 수 없다. 따라서 4팀의 2 9Round경기도 서로 바꾸어 준다.

이것은 Swap Rounds의 일반화된 형태이다.

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	2	3	-5	-4	-3	5	4	-6
2	5	1	-1	-5	4	3	6	-4	-6	-3
3	-4	5	4	-1	6	-2	1	-6	-5	2
4	3	6	-3	-6	-2	1	5	2	-1	-5
5	-2	-3	6	2	1	-6	-4	-1	3	4
6	-1	-4	-5	4	-3	5	-2	3	2	1

  

T\R	1	2	3	4	5	6	7	8	9	10
1	6	4	2	3	-5	-4	-3	5	-2	-6
2	5	-6	-1	-5	4	3	6	-4	1	-3
3	-4	5	4	-1	6	-2	1	-6	-5	2
4	3	6	-1	-3	-6	-2	1	5	2	-5
5	-2	-3	6	2	1	-6	-4	-1	3	4
6	-1	2	-5	4	-3	5	-2	3	-4	1

Figure 5. 2팀 2, 9Round의 Partial Swap Rounds

5. Partial Swap Teams

두개의 팀을 선택하고, 특정 Round를 선택하여 특정 라운드 두 팀의 경기를 서로 바꾸어준다. 예를 들어 2, 4팀을 선택하고, 9Round를 선택하면 2팀의 본래 9Round의 경기 상대인 1팀이 4팀과 경기를 하게 되고 4팀의 경기 상대인 1팀이 2팀과 경기를 하게 된다.

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	4	3	6	-4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	-2	1	5	2	-6	-3
5	-2	-3	6	4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

  

T\R	1	2	3	4	5	6	7	8	9	10
1	6	-2	2	3	-5	-4	-3	5	4	-6
2	5	1	-1	-5	4	3	6	-4	-6	-3
3	-4	5	4	-1	6	-2	1	-6	-5	2
4	3	6	-3	-6	-2	1	5	2	-1	-5
5	-2	-3	6	2	1	-6	-4	-1	3	4
6	-1	-4	-5	4	-3	5	-2	3	2	1

Figure 6. 2, 4팀의 9Round Partial Swap Rounds

한국 프로야구 리그는 TTP의 확장된 형태이다. 본 연구에서는 한국 프로야구 리그의 42Round를 각14Round 3개로 분할하여 각각을 TTSA를 응용하여 TTP 일정을 만들어 낸 후 이를 결합하여 일정을 수립하였다. 3개의 일정을 결합하여 만든 42Round의 일정이 제약조건을 만족시키지 못할 경우에는 일반적인 SA 알고리즘을 사용하여, 해를 찾는 과정을 반복한다. 이때는 해의 탐색영역을 가능해로 제한하여 해를 찾는다.

4. 실험결과

본 연구에서는 시뮬레이터를 작성하여 실험을 하였다. 시뮬레이터에 설정한 parameter 값은 <Table 2>와 같다.

Table 2. 시뮬레이터의 Parameter 설정

counterLimit	5
maxPhase	2000
maxReheat	3
$\omega$	1.3
$\Theta$	1.0003
$\delta$	1.0003
cooling Rate	0.999
Initial Temperature	8000

실제 2005년도 한국 프로야구 리그에는 제약조건을 위반하는 부분이 존재한다. 특정 팀이 원정 9경기를 연속으로 치르도록 일정이 편성되어 있다. 물론 그 사이에 휴식일이 있으나, 이것은 다른 팀과의 형평성에서 문제가 될 수도 있다.

본 연구에서 수행한 실험에서는 앞서 기술한 제약조건을 모두 만족하면서도 각 목적함수에 대하여 <Table 3>과 같이 개선된 일정을 구할 수 있었다.

Table 3. 시뮬레이터 실험결과 비교(단위 km)

	기존일정	실험결과	개선율
이동거리 총합	95511.0	81968.0	14.2%
이동거리 차이의 합	142777.0	106372.0	25.5%
최대 이동거리	20925.0	16497.0	21.2%

본 실험을 수행한 환경은 Pentium4 2.0 CPU, 512 RAM의 Windows 환경의 컴퓨터에서 Java로 작성된 Simulator를 사용하여 실험을 하였다. 계산시간은 3개의 TTP로 해를 구한 계산 시간을 합한 것이다.

Table 4. 시뮬레이터 계산시간(단위 초)

	계산시간
이동거리 총합	30262.6
이동거리 차이의 합	28134.3
최대 이동거리	26172.8

5. 결론

기존에 연구된 SLSP의 방법론을 적용하여 한국 프로야구 리그 일정을 수립한 결과 목적함수 별로 14%~22% 정도 더 우수한 일정을 구할 수가 있었다. 이는 한국 프로야구 리그뿐만 아니라 여타 다른 프로 스포츠 리그에도 적용할 수 있음을 보여준다. 그러나 현재의 방법론은 프로리그에서 요구하는 모든 제약조건을 포함하기에는 어려운 점이 있다. 앞으로 방법론적으로 이러한 부분을 개선하여 프로리그의 복잡한 요구사항화 제약조건을 가지는 만족시면서 효율적인 일정을 수립할 수 있도록 해야할 것이다.

참고문헌

McAloon K., Tretkoff C. and Wetzel G.(1997), Sports League Scheduling, In Proceedings of the third ILOG Optimization Suite International Users' Conference  
Easton K., Nemhauser G. and Trick M.(2001), The Traveling Tournament Problem Description and Benchmarks, In Proceedings of the 7th International Conference on Principles and Practice of Constraint

2005 한국경영과학회/대한산업공학회 춘계공동학술대회  
2005년 5월 13일~14일, 충북대학교

- Programming, 580-584.
- Charles F. and Jacques A. F.(1993), Allocating Games for the NHL Using Integer Programming, Operations Research 41(4), 649-654
- T. Benoist, F. Laburthe and B. Rottembourg(2001), Lagrange Relaxation and Constraint Programming Collaborative Schemes for Traveling Tournament Problem, CPAI-OR, Wye College, 15-26
- Easton K., Nemhauser G. and Trick M.(2003), Solving the Traveling Tournament Problem : A Combined Integer Programming and Constraint Programming Approach, PATAT 2002, LNCS 2740, 100-109
- A. Anagnostopoulos, L. Michel, V. Hentzenryck and Y. Vergados(2003), A Simulated Annealing Approach to the Traveling Tournament Problem, CPAI-OR
- Cynthia B., Ellis L. J., George L. N., Martin W. P. S. and Pamela H. V.(1998), Branch-and-Price: Column Generation for Solving Huge Integer Programs, Operations Research 46(3), 316-329