IME639: Course Project

Sports Timetabling Problem

- **Anujendra (20114262)**
- **Dhanish Kumar (170243)**
- **Shayan Ul Haq (180706)**

Sports Timetabling Problem

The main purpose of any sports scheduling problem is to find a time slot for each game based on some predefined constraints. The tournament can be of two types i.e., single league or multi-league. There are different ways in which it can be scheduled. It can be scheduled as knockout tournament where the winner of one round advances to the next round and the loser is eliminated. It can also be scheduled as a round robin tournament where each team plays with every other team. The teams can meet any number of times in this type of tournament but the number of games should be same for all the teams.

In this problem, a single league, double round robin tournament is considered. In a double round robin tournament, each team plays the other teams twice. These games are played as home and away i.e., each team is host for one game and visitor in the other. The problem is time constraint and should complete in minimum number of time slots. In this scenario, the total number of slots would be equal to the total number of games played by a particular team. Fig 1 shows a schedule of 6 team and 10 slots in 2RR tournament type.

```
Round 0: (1 , 3) (2 , 5) (4 , 0)
Round 1: (2 , 1) (3 , 4) (5 , 0)
Round 2: (0 , 1) (2 , 3) (4 , 5)
Round 3: (1 , 2) (3 , 0) (5 , 4)
Round 4: (0 , 3) (4 , 1) (5 , 2)
Round 5: (0 , 5) (2 , 4) (3 , 1)
Round 6: (1 , 0) (4 , 2) (5 , 3)
Round 7: (0 , 4) (1 , 5) (3 , 2)
Round 8: (1 , 4) (2 , 0) (3 , 5)
Round 9: (0 , 2) (4 , 3) (5 , 1)
```

Figure 1: Sample schedule for n=6 teams and 2(n-1) = 10 slots*

The whole season of a double round robin tournament is split into two equally long intervals. These intervals can be seen as two different 1RR intervals, in which the home- away status of mutual games of two teams alternates between successive intervals. The schedule which follows this property is termed as ‘phased’.

Apart from these, there are two other types of constraints i.e., hard constraints and soft constrains. The hard constraints are basic property of timetable and it cannot be violated whereas soft constraints are preferences that should be followed whenever possible. If a soft constraint is violated, a penalty will be added to the objective. The primary goal of the problem is to minimize the total penalties incurred.

Standardized RobinX XML data is used for problem instances. The format enhances problem data sharing and reuse among different users. It minimizes problem specification part which is the hardest task and increases the accessibility.

Constraints:

1. Capacity Constraint 1: It is the “place constraint” which stops a team to play a home game or away game in a specific time slot. It’s a hard constraint
2. Capacity Constraint 2: It is the “top team and bottom team constraint” which stops teams at the bottom to play all initial games against the teams at the top. It’s a soft constraint.
3. Capacity Constraint 3: It is the limitation on the length of home stands by not allowing successive home breaks. It’s a soft constraint.
4. Capacity Constraint 4: It puts limitation on the total number of games between top teams or it can even put a limit on the total number of home games per slot. It’s a hard constraint.
5. Game Constraint 1: It can restrict or schedule a game to a time slot. In real world, these constraints turn out to be helpful when a broadcaster might ask for top game in a particular slot. It’s a hard constraint.
6. Break Constraint 1: It is used to restrict total number of breaks per team or can even avoid breaks at the starting or closing of the season. It’s a hard constraint.
7. Break Constraint 2: It is used to restrict the total number of breaks for a subset of teams. It’s a hard constraint.
8. Fairness Constraint 2: It makes sure that the difference in played home games between any two teams should be smaller than intp at any particular time. It’s a hard constraint.
9. Separation Constraint 1: It makes sure that two games between the same opponent should be separated by at least a particular number of slots. It’s a hard constraint.

Sports Timetabling

We will organize the sports timetabling using the double round robin process.

Say, n = total teams.

Decision Variable, $x_{ijk} = \begin{cases} 1, & \text{if } i^{th} \text{ team plays home game against the } j^{th} \text{ team (home away) in the } k^{th} \text{ slot} \\ 0, & \text{otherwise} \end{cases}$

where $i, j \in \{0, 1, 2, \dots, n-1\}$

$k \in \{0, 1, 2, \dots, 2(n-1)\}$

Basic Constrain

i) No team plays against itself $\Rightarrow$ ~~x_{ij}~~ $x_{iik} = 0 \quad \forall i, k$

ii) Each team plays exactly once in a single round $\Rightarrow \sum_j' (x_{ijk} + x_{jik}) = 1$

iii) Each team plays exactly $(n-1)$ home games $\Rightarrow \sum_k \sum_j x_{ijk} = n-1$

iv) Each team plays 1 home and ~~as~~ 1 away against each other. $\Rightarrow \sum_k' (x_{ijk}) = 1$

Objective function

minimize $\sum' c_i p_i$ for every soft constrain 'i'.

Phase Constant ⇒ If there is a phased constant i.e. home game = P then according to double round robin tournament it is split into two equally long single round robin tournament for which home-away status of mutual game for each pair of teams alternates b/w consecutive intermode then s.t

$$\sum_{K=1}^{n-1} (x_{ijK} + x_{jiK}) = 1 \quad , \quad \forall i,j \text{ s.t } i \neq j$$

1) Hard Constrains

i) CA_1 , where i = team

K = slots

$j \in \{0, 1, \ldots, n-1\}$

if mode = H $\quad \sum_j \sum_K x_{ijK} \leq max.$

if mode = A $\quad \sum_j \sum_K x_{jiK} \leq max.$

ii) CA_2 , where i = team 1

$j \in$ team 2

$K \in$ slots

if mode 1 = H $\quad \sum_j \sum_K x_{ijK} \leq max \quad \forall i \in$ team 1

if mode 1 = A $\quad \sum_j \sum_K x_{jiK} \leq max. \quad \forall i \in$ team 1

if mode 2 = HA $\quad \sum_j \sum_K (x_{ijK} + x_{jiK}) \leq max.$

iii) CA_3, where $i \in team1$

$j \in team2$

$l \in \{0, 1, 2, \ldots, n - intp\}$

$k \in \{l, l+1, \ldots, l + intp - 1\}$

if mode 1 = H $\quad \sum_k \sum_j x_{ijk} \leq max. \quad \forall l \in \{0, 1, 2, \ldots, n - intp\}$

$\forall i \in \{ team1$

if mode 1 = A $\quad \sum_k \sum_j x_{jik} \leq max. \quad \forall l \in \{0, 1, 2, \ldots, n - intp\}$

$\forall i \in team1$

if mode = HA $\quad \sum_k \sum_j (x_{ijk} + x_{jik}) \leq max. \quad \forall l, \forall i \in team1$

iv) CA_4, where $i \in team1$

$j \in team2$

$k \in slots$

mode 2 = Global $\quad$ mode 2 =

mode1 = H $\quad \sum_i \sum_j \sum_k x_{ijk} \leq max \quad\quad \sum_i \sum_j x_{ijk} \leq max \quad \forall k$

mode 2 = A $\quad \sum_i \sum_j \sum_k x_{ijk} \leq max \quad\quad \sum_i \sum_j x_{jik} \leq max \quad \forall k$

mode 3 = HA $\quad \sum_i \sum_j \sum_k (x_{ijk} + x_{jik}) \leq max \quad\quad \sum_i \sum_j (x_{ijk} + x_{jik}) \leq max \quad \forall k$

Game Constraints

GA_1, where $s \in \{0,1,2,\ldots, \text{meeting-size}-1\}$

meetings $\in \{i_1,j_1;\ i_2,j_2;\ i_3,j_3;\ \ldots;\ i_s,j_s\}$

$k \in$ slots

$$\min \le \sum_s \sum_k x_{i_s j_s k} \le \max$$

Break Constraints

i) BR_1, where i = teams, $j \in \{0,1,2,\ldots,n-1\}$

k = slots, $l \in \{0,1,2,\ldots,n-1\}$

if there is a home break in k^{th} slot for team 'i', then
$$\left(\sum_j x_{ijk-1}\right)\left(\sum_j x_{ijk}\right) = 1 \quad , \text{mode2} = H$$

if there is a away break in k^{th} slot for team 'i', then
$$\left(\sum_j x_{jik-1}\right)\left(\sum_j x_{jik}\right) = 1 \quad , \text{mode2} = A$$

if there is a home-away break in k^{th} slot for team 'i' then
$$\left(\sum_j x_{ijk-1} \sum_j x_{ijk} + \sum_j x_{jik-1} \sum_j x_{jik}\right) = 1 \quad , \text{mode2} = HA$$

so, the constraint would be given as

mode 2 = AH, $\sum_k \sum_j x_{ijk-1} \sum_l x_{ilk} \le intp$

$$\sum_k \sum_j \sum_l x_{ijk-1}\, x_{ilk} \le intp$$

mode 2 = A, $\sum_k \sum_l \sum_j x_{jik-1}\, x_{lik} \le intp$

$$\sum_k \sum_l \sum_j \left(x_{ijk-1}\, x_{ilk} + x_{jik-1} \cdot x_{lik}\right) \le intp$$

ii) BR_2 , where $i \in teams$

$j \in \{0,1,2,\dots,n-1\}$

$l \in \{0,1,2,\dots,n-1\}$

$k \in slots$

$$\sum_i \sum_j \sum_k \sum_l \left(x_{ijk-1} \cdot x_{ilk} + x_{jik-1} \cdot x_{lik}\right) \leq intp$$

Fairness Constraints

FA , where $i \in teams$, $j \in \{0,1,2,\dots,n-1\}$

$l \in teams$, $k \in \{ slots$

Number of home game of i^{th} team after k^{th} time slot $\Rightarrow \sum_i \sum_j \sum_k x_{ijk}$

So,

$$-intp \leq \sum_i \sum_j \sum_k x_{ijk} - \sum_l \sum_j \sum_k x_{ljk} \leq intp \quad \forall \ (i,l) \in teams \times teams$$

Seperation Constraint

SE_1 , where $i,j \in teams$

$k \in \{0,1,2,\dots,2n-1\}$

$$\sum_k (k \cdot x_{ijk} - k \cdot x_{jik}) \geq (min+1) \ , \forall i,j \in teams \text{ s.t } i \neq j$$

or

$$\sum_k (k \, x_{ijk} - k \cdot x_{jik}) \leq -(min+1) \ , \forall i,j \in teams \text{ s.t } i \neq j$$

2) Soft Constraint

let $\sum_i \sum_j \sum_k x_{ijk} \leq \text{intp}$

be the crossponding hard constraint then for the soft constrain, we define

$$\text{Deviation } (d_i) = \max\left(0, \sum_i \sum_j \sum_k x_{ijk} - \text{intp}\right)$$

and for implementation we will use 2 constraint

i) $d \geq 0$

ii) $d \geq \sum_i \sum_j \sum_k x_{ijk} - \text{intp}$

3) Objective

Objective function $= \sum P_i d_i$, $i \in$ soft constraint

P_i = ~~primary~~ penalty associated with soft constraint i

d_i = deviation calculated

So,

objective : $\min \sum P_i d_i$

Complexity Analysis

- **Decision Variables** : (6<=(n =no. of teams)<=20)
  - **x[i][j][k]** -> **Total count**: n*n*(2(n-1)) (15,200 in worst case!) ->**Type:** binary
  - **h[i][k]** -> **Total count:** n*(2(n-1))(760 in worst case)-> **Type:** binary
  - **a[i][k]**-> **Total count:** n*(2(n-1)) (760 in worst case)-> **Type:** binary
  - **d[i]** -> **Total count:** no. of soft constraints (O(10,000))-> **Type:** positive integer
  - **y[i]-> Total count:** no. of separation constraints (O(10,000)) ->**Type:** binary

  Hence, Total decision variables are O(10,000) (may reach as high as 1,00,000)

- **Constraints**:
  - Basic constraints: O(10000)
  - Capacity constraint : (O(100) such constraints ,each one of them belongs to either of the below constraints)
    - 1 CA1 => 1 constraint
    - 1 CA2 => 1 constraint
    - 1 CA3 =>O(n*n) constraints
    - 1 CA4=>O(n) constraints
  - Game Constraints:(O(10))
    - 1GA1=>1 constraint
  - Break Constraint:(O(10))
    - 1BR1=>1 constraint
    - 1BR2=>1constraint
  - Fairness Constraint: (O(10))
    - 1FA2=>(O(n*n*n))constraints
  - Separation constraints : (O(10))
    - 1SE1=>O(n*n)constraints

  Hence, total individual constraints can be higher than 1,00,000!

- Evaluation of a single constraint can reach as high as O(n^5) , on an average O(n^2)
- From above it is clear that the complexity of program depends on:
  - n
  - no. of constraints in input file
  -
- **Owing to above factors run time for the test cases with higher constraints or larger n is higher.**
- **(For eg, "ITC2021_Test5.xml" with n= 16, took around 35 minutes)**