DM204, 2011
SCHEDULING, TIMETABLING AND ROUTING

Discussion session
Sport Timetabling

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Overview

Problems we treat:

- single and double round-robin tournaments
- balanced tournaments
- bipartite tournaments

Solutions:

- general results
- graph algorithms
- integer programming
- constraint programming
- metaheuristics

Outline

1. Problem Definition and Combinatorial Design Approach

2. Decomposition Approach

Terminology

- A schedule is a mapping of games to slots or time periods, such that each team plays at most once in each slot.
- A schedule is compact if it has the minimum number of slots.
- Mirrored schedule: games in the first half of the schedule are repeated in the same order in the second half (with venues reversed)
- Partially mirrored schedule: all slots in the schedule are paired such that one is the mirror of the other
- A pattern is a vector of home (H) away (A) or bye (B) for a single team over the slots
- Two patterns are complementary if in every slot one pattern has a home and the other has an away.
- A pattern set is a collection of patterns, one for each team
- A tour is the schedule for a single team, a trip a series of consecutive away games and a home stand a series of consecutive home games

Round Robin Tournaments

(round-robin principle known from other fields, where each person takes an equal share of something in turn)

- Single round robin tournament (SRRT) each team meets each other team once
- Double round robin tournament (DRRT) each meets each other team twice

Definition SRRT Problem

Input: A set of n teams $T = \{1, \dots, n\}$

Output: A mapping of the games in the set $G = \{g_{ij} : i, j \in T, i < j\}$, to the slots in the set $S = \{s_k, k = 1, \dots, n-1$ if n is even and $k = 1, \dots, n$ if n is odd$\}$ such that no more than one game including i is mapped to any given slot for all $i \in T$.

Circle method

Label teams and play:

```
Round 1. (1 plays 14, 2 plays 13, ... )
 1  2  3  4  5  6  7
 14 13 12 11 10 9  8
```

Fix one team (number one in this example) and rotate the others clockwise:

```
Round 2. (1 plays 13, 14 plays 12, ... )
 1  14 2  3  4  5  6
 13 12 11 10 9  8  7

Round 3. (1 plays 12, 13 plays 11, ... )
 1  13 14 2  3  4  5
 12 11 10 9  8  7  6
```

Repeat until almost back at the initial position

```
Round 13. (1 plays 2, 3 plays 14, ... )
 1  3  4  5  6  7  8
 2 14  13 12 11 10 9
```

Definition DRRT Problem

Input: A set of n teams $T = \{1, \ldots, n\}$.

Output: A mapping of the games in the set $G = \{g_{ij} : i, j \in T, i \neq j\}$, to the slots in the set $S = \{s_k, k = 1, \ldots, 2(n-1)$ if n is even and $k = 1, \ldots, 2n$ if n is odd$\}$ such that no more than one game including i is mapped to any given slot for all $i \in T$.

The schedule can be obtained by the circle method plus mirroring

Venue assignment can also be done through the circle method

Latin square

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 5 & 1 & 4 \\ 3 & 5 & 4 & 2 & 1 \\ 4 & 1 & 2 & 5 & 3 \\ 5 & 4 & 1 & 3 & 2 \end{bmatrix}$$

alldiff in rows and columns
easy to generate: permutation on first row and then cyclic permutations

Interested in even, symmetric Latin square with all identical l_{ii}, $i = 1..n$
$\rightsquigarrow$ SRRT

Example: 4 Teams

```
round 1: 1 plays 2, 3 plays 4
round 2: 2 plays 3, 1 plays 4
round 3: 3 plays 1, 2 plays 4
```

Rewrite the schedule as a multiplication table: "a plays b in round c".

```
     1  2  3  4
    -----------
1 |     1  3  2
2 | 1      2  3
3 | 3  2      1
4 | 2  3  1
```

If the blank entries were filled with the symbol 4, then we have an even, symmetric latin square.

Round robin tournaments with preassignments correspond to complete partial latin squares ➜ NP-complete

Inverse approach:

- determining the venue for each game
- assigning actual teams to games

Decomposition:

1. First generate a pattern set
2. Then find a compatible pairing place holders–games (this yields a timetable)
3. Then assign actual teams to the place holders

More on this later

Generation of feasible pattern sets

- In SRRT:
  - every pair of patterns must differ in at least one slot. $\Rightarrow$ no two patterns are equal in the pattern set
  - if at most one break per team, then a feasible pattern must have the complementary property ($m = n/2$ disjoint, complementary pairs of patterns)
- In DRRT,
  - for every pair of patterns i, j such that $1 \leq i < j \leq n$ there must be at least one slot in which i is home and j is away and at least one slot in which j is at home and i is away.
  - every slot in the pattern set includes an equal number of home and away games.

Definition Balanced Tournament Designs (BTDP)

Input: A set of n teams $T = \{1, \ldots, n\}$ and a set of facilities F.

Output: A mapping of the games in the set $G = \{g_{ij} : i, j \in T, i < j\}$, to the slots available at each facility described by the set $S = \{s_{fk}, f = 1, \ldots, |F|,\ k = 1, \ldots, n-1$ if n is even and $k = 1, \ldots, n$ if n is odd$\}$ such that no more than one game involving team i is assigned to a particular slot and the difference between the number of appearances of team i at two separate facilities is no more than 1.

- BTDP($2m$,m): $2m$ teams and m facilities. There exists a solution for every $m \neq 2$.

- BTDP($2m+1, m$): extension of the circle method:

  Step 1: arrange the teams $1, \ldots, 2m+1$ in an elongated pentagon. Indicate a facility associated with each row containing two teams.

  Step 2: For each slot $k = 1, \ldots, 2m+1$, give the team at the top of the pentagon the bye. For each row with two teams i, j associated with facility f assign g_{ij} to s_{kf}. Then shift the teams around the pentagon one position in a clockwise direction.

Bipartite Tournament

Input: Two teams with n players $T_1 = \{x_1, \ldots, x_2\}$ and $T_2 = \{y_1, \ldots, y_n\}$.

Output: A mapping of the games in the set $G = \{g_{ij}\ i \in T_1, j \in T_2\}$, to the slots in the set $S = \{s_k,\ k = 1, \ldots, n\}$ such that exactly one game including t is mapped to any given slot for all $t \in T_1 \cup T_2$

Latin square $\Leftrightarrow$ bipartite tournament ($l[i, j]$ if player x_i meets player y_j in l_{ij})

Extensions:

- n facilities and seek for a balanced BT in which each player plays exactly once in each facility $\iff$ two mutually orthogonal Latin squares (rows are slots and columns facilities)

  A pair of Latin squares $A = [a_{ij}]$ and $B = [b_{ij}]$ are orthogonal iff the the ordered pairs (a_{ij}, b_{ij}) are distinct for all i and j.

1	2	3		1	2	3		1 1	2 2	3 3
2	3	1		3	1	2		2 3	3 1	1 2
3	1	2		2	3	1		3 2	1 3	2 1
	A				B				A and B superimposed	

  Mutually orthogonal Latin squares do not exist if $m = 2, 6$.

- Chess tournaments (assigning white and black)

- avoid carry-over effects, no two players x_i and y_j may play q the same sequence of opponents y_p and followed immediately by y_q. $\Rightarrow$ complete latin square.

Graph Algorithms

A spanning subgraph of $G = (V, E)$ with all vertices of degree k is called a k-factor (A subgraph $H \subseteq G$ is a 1-factor $\Leftrightarrow$ $E(H)$ is a matching of V)

A 1-factorization of $K_n \equiv$ decomposition of K_n in perfect matchings $\equiv$ edge coloring of K_n

A SRRT among $2m$ teams is modeled by a complete graph K_{2m} with edge (i, j) representing the game between i and j and the schedule correspond to an edge coloring.

To include venues, the graph K_{2m} is oriented
(arc (ij) represents the game team i at team j)
and the edge coloring is said an oriented coloring.

A DRRT is modeled by the oriented graph G_{2m} with two arcs a_{ij} and a_{ji} for each ij and the schedule correspond to a decomposition of the arc set that is equivalent to the union of two oriented colorings of K_{2m}.

Assigning venues with minimal number of breaks:

- SRRT: there are at least $2m - 2$ breaks. Extension of circle method.
- DRRT: Any decomposition of G_{2m} has at least $6m - 6$ breaks.
- SRRT for n odd: the complete graph on an odd number of nodes K_{2m+1} has an oriented factorization with no breaks.

Outline

1. Problem Definition and Combinatorial Design Approach

2. Decomposition Approach

Three phase approach by IP

1. Generate a timetable, ie, a schedule for games (basic SRRT)

   Variable

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j = 1, \ldots, n;\ i < j,\ k = 1, \ldots, n-1$$

   Every team plays exactly once in each slot

$$\sum_{j : j > i} x_{ijk} = 1 \qquad \forall i = 1, \ldots, n;\ k = 1, \ldots, n-1$$

   Each team plays every opponent exactly once.

$$\sum_{k} x_{ijk} = 1 \qquad \forall i, j = 1, \ldots, n;\ i < j$$

Branch and cut algorithm
Adds odd-set constrains that strengthen the one-factor constraint, that is, exactly one game for each team in each slot

$$\sum_{i \in S, j \notin S} x_{ijk} \leq 1 \qquad \forall S \subseteq T, |S| \text{ is odd, } k = 1, \ldots, n-1$$

2. Enumerate all possible patterns and put them in S. Then select among them with the following IP:

$$x_i = \{0, 1\} \qquad \forall i \in S$$

h_{ik} is 1 if patter i scheduled has a H at k, m number of games per round:

$$\sum_{i \in S} x_i h_{ik} = m \qquad \forall\, k \in T$$

a_{ik} is 1 if patter i scheduled has a A at k:

$$\sum_{i \in S} x_i a_{ik} = m \qquad \forall\, k \in T$$

c_i cost of pattern i determined, eg, by number of breaks

$$\min \sum_{i \in S} c_i x_i$$

CP formulation

- CP for phase 1 (games and patterns)

```
int n = ...;
range Teams [1..n];
range Slots [1..n-1];
var Teams opponent[Teams,Slots];

solve {

   forall (i in Teams, k in Slots) opponent[i,t]<>i;
   forall (i in Teams) alldifferent(all (k in Slots) opponent[i,k]);
   forall (k in Slots) onefactor(all (i in Teams) opponent[i,k]);

};
```

- CP for phase 2: assign actual teams to position in timetable

Constraints to be included in practice:

- Pattern set constraints
  - feasible pattern sequences: avoid three consecutive home or away games
  - equally distributed home and away games

- Team-specific constraints
  - fixed home and away patterns
  - fixed games and opponent constraints
  - stadium availability
  - forbidden patterns for sets of teams
  - constraints on the positioning of top games

Objective: maximize the number of good slots, that is, slots with popular match-ups later in the season or other TV broadcasting preferences.

Application Examples

- Dutch Professional Football League [Schreuder, 1992]
  1. SRRT canonical schedule with minimum breaks
     and mirroring to make a DRRT
  2. assign actual teams to the patterns

- European Soccer League [Bartsch, Drexl, Kroger (BDK), 2002]
  1. DRRT schedule made of two separate SRRT with complementary patterns (Germany)
     four SRRTs the (2nd,3rd) and (1st,4th) complementary (Austria)
  2. teams assigned to patterns with truncated branch and bound
  3. games in each round are assigned to days of the week by greedy and local search algorithms

- Italian Football League [Della Croce, Olivieri, 2006]
  1. Search for feasible pattern sets appealing to TV requirements
  2. Search for feasible calendars
  3. Matching teams to patterns

Reference

- Kelly Easton and George Nemhauser and Michael Trick, Sport Scheduling, in Handbook of Scheduling: Algorithms, Models, and Performance Analysis, J.Y-T. Leung (Ed.), Computer & Information Science Series, Chapman & Hall/CRC, 2004.