Sudoku Team Pairing

James Taylor

August 9, 2021

1 The problem

Suppose we have 6 teams and want each team to play every other team over 5 weeks. We don't want to repeat any matches, and we want each team to be playing for every week. A match between teams i and j will be represented by the "match tuple" (i, j).

There are two conditions we need to satisfy when filling out the roster, which might look like this:

Week 1: $(1, 2), (3, 4), (5, 6)$

Week 2: $(1, 3), (2, 5), (4, 6)$

Week 3: $\dots$

The first condition is that any given number in a pair does not show up in any other pairs (because a team can only play against one other team at a time). The second condition is that a pair in a given week does not show up in a previous week.

We can represent the team pairings for a given week in a matrix, W. Putting a 1 in row i and column j means team i plays against team j. So if we have team pairing (i, j), then $W(i, j) = 1$. A zero means those teams do not play against each other. The diagonals must be 0 because a team cannot play against itself. So the example in the first week above is represented as:

j

		1	**2**	**3**	**4**	**5**	**6**
	1	0	1	0	0	0	0
	2	1	0	0	0	0	0
i	**3**	0	0	0	1	0	0
	4	0	0	1	0	0	0
	5	0	0	0	0	0	1
	6	0	0	0	0	1	0

The matrix is symmetric because team-pairing (i, j) is equivalent to (j, i). When filling out the matrix for a given week, the first condition can be satisfied by making sure that, when deciding to put a 1 in a given position, (i, j), there are no other 1's in that row and column. So each row or column may only contain a single 1. If the row contains another 1, that's equivalent to having the pairing (i, k) or (k, i) somewhere on that week's roster. If the column contains another 1, that's equivalent to already having the pairing (j, k) or (k, j) somewhere.

Having the matrix be symmetric rather than just filling out one half is what allows it to also check (k, i) and not just (i, k).

In order to fill out the roster for multiple weeks, we simply put the week number rather than just 1. Having 3 in the $(2, 3)$ position means team 2 plays against team 3 on week 3. We modify the rule about not having any other 1's to just be that when putting a given week number in a certain position, that week number must not show up in the same row and column. The second condition that a given pair cannot show up in a previous week is automatically satisfied since a given position in the matrix can only have a single week number.

The practical result of these conditions is that each row and column will not contain any repeat digits. In order for each team to play each other team, we must have a full matrix (every team must be paired). In order to have a full roster (no idle teams in a given week), then every column/row must contain each week number.

These rules are equivalent to playing sudoku with the digits 0 to $n - 1$, except we have the additional constraint that the board must be symmetric and the diagonals are 0, and we do not have any box constraints.

The number of possible pairings is given by $\frac{n(n-1)}{2}$. Assuming n is even, the number of pairings in a week is $\frac{n}{2}$. Therefore, the minimum number of weeks necessary for every team to play against every other team is

$$\frac{\frac{n(n-1)}{2}}{\frac{n}{2}} = n - 1$$

This assumes it is possible to fill the sudoku matrix. If it is not possible, additional weeks must be added and some teams will not be playing on those weeks.

Here is an algorithm that worked for me when filling out the sudoku matrix for 6 teams over 5 weeks. No guarantees it is a generally correct algorithm.

Fill the first row with 1 to $n - 1$. Remember to fill the corresponding column out the same when filling out a row.

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	3	4	5
2	1	0				
3	2		0			
4	3			0		
5	4				0	
6	5					0

Let k be the number you are considering filling in for a given position. When starting a new row, for the first position, set $k = b + 1$ where b is the number you filled in for the first position in the previous row. If k does not show up anywhere else in the row or column, choose that number. Otherwise, increment until you find one that works. If you pass $n - 1$, wrap back to 1.

When filling the rest of the row, increment the k value you chose for the previous position and continue the procedure. Following this algorithm yielded this completed sudoku:

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	2	3	4	5
2	1	0	3	4	5	2
3	2	3	0	5	1	4
4	3	4	5	0	2	1
5	4	5	1	2	0	3
6	5	2	4	1	3	0

2 Linear Algebra Perspective

We can represent the roster matrix R as the sum of the $n \times n$ matrices for each week W_i:

$$R = W_1 + 2W_2 + 3W_3 + \ldots + (n-1)W_{n-1}$$

The conditions on the matrix W_i are that each row and column contains zeros and a single 1 and that

$$W_i^T = W_i$$

Additionally, the diagonals must be zero. This can be formulated as $\mathrm{tr}(W_i) = 0$, which implies the eigenvalues must sum to 0.

In order to get the condition that the elements of the matrices W_i and W_j must not overlap, we can ensure that the dot product of a given row of one of the matrices with the same row of the other matrix is zero. Because the matrices are symmetric, this is true if their matrix product has diagonals of 0:

$$\mathrm{tr}(W_i W_j) = 0 \quad \forall i \neq j$$

Thus, the eigenvalues also sum to zero for the products of these matrices.

Each matrix is a symmetric orthonormal matrix:

$$W_i^2 = I$$

This is because multiplying the matrix by itself creates a dot product of unit vectors with themselves for the diagonals. The non-diagonals are 0 because each column is orthogonal to all the others.