Berger explained

Otto Milvang, September 21, 2024

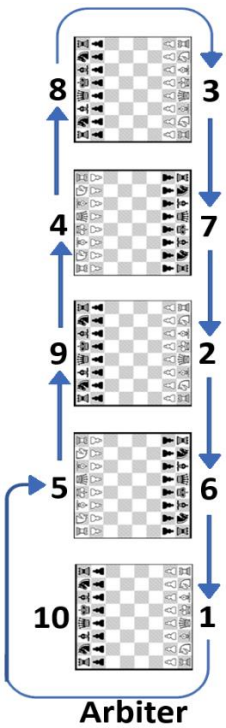
Introduction

The Berger tables is a predefined pairing system used for Round Robin tournaments. The Berger tables are described in FIDE Handbook C. 05, Annex 1 http://handbook.fide.com/chapter/C05Annex1: Details of Berger Table. This article shows an efficient way to use them in round robin tournaments, and how to calculate the Berger tables. The document describes only Berger for even number of players. Where there is an odd number of players, the highest number counts as a bye. In all example we will show Berger tables for 10 players.

Berger tables 10 players

Rd 1:	1 - 10	2 - 9	3 - 8	4 - 7	5 - 6
Rd 2:	10 - 6	7 - 5	8 - 4	9 - 3	1 - 2
Rd 3:	2 - 10	3 - 1	4 - 9	5 - 8	6 - 7
Rd 4:	10 - 7	8 - 6	9 - 5	1 - 4	2 - 3
Rd 5:	3 - 10	4 - 2	5 - 1	6 - 9	7 - 8
Rd 6:	10 - 8	9 - 7	1 - 6	2 - 5	3 - 4
Rd 7:	4 - 10	5 - 3	6 - 2	7 - 1	8 - 9
Rd 8:	10 - 9	1 - 8	2 - 7	3 - 6	4 - 5
Rd 9:	5 - 10	6 - 4	7 - 3	8 - 2	9 - 1

How to run a Round Robin tournament based on Berger tables



The players sit randomly or after drawing of lots. The boards are ordered every other white/black, as in the figure, always with the board closest to the arbiter with black to the left of the arbiter. The player sitting here has the highest start number and shall sit on the same place throughout the tournament and thus called the fixed player.

The number of the players counterclockwise from the arbiter follows the order of players meeting the player with the highest start number. Another approach is to place the players counterclockwise first the white players from 1 to N/2 (where N is the number of players), and then the black players from N/2+1 to N.

When the 1st round is finished, all players move one step clockwise around the table but skip over the fixed player. The fixed player shall turns the board after each round, so he alternate black and white games. This is how the tournament continues until everyone has met everyone.

In case of odd number of players, the board next to the arbiter can be removed, just remember that player 1 is the player that will have a bye in round 1.

This scheme will follow the Berger tables. The board numbers follow the white players start number from round 1.

Result tables

There are mainly two types of result tables for Berger tournaments, the cross table and the Berger result table.

	Player	1	2	3	4	5	6	7	8	9	10	Total	Position
1		X											
2			X										
3				X									
4					X								
5						X							
6							X						
7								X					
8									X				
9										X			
10											X		

The cross table is easy to understand but will not show which round a game was played nor which colors they played.

Start no:	Name: Round:	1	2	3	4	5	6	7	8	9	Points	Tie break	Place
1		10	2	3	4	5	6	7	8	9			
2		9	1	10	3	4	5	6	7	8			
3		8	9	1	2	10	4	5	6	7			
4		7	8	9	1	2	3	10	5	6			
5		6	7	8	9	1	2	3	4	10			
6		5	10	7	8	9	1	2	3	4			
7		4	5	6	10	8	9	1	2	3			
8		3	4	5	6	7	10	9	1	2			
9		2	3	4	5	6	7	8	10	1			
10		1	6	2	7	3	8	4	9	5			

Start no. 10 plays with black in round 1

The Berger result table has one column per round and will also give information about the colors. For example, player 3 in round 7 has the number 5 in the left corner. Then player 5 has white against player 3, 5-3.

Create your own Berger tables at: https://www.milvang.no/spp/berger/berger.html

How to build the Berger tables

Let N be the number of players (or number of players + 1 if the number of players are odd). The players are assigned a pairing number 1 to N. The number of boards is B = N/2. In the first round the lowest half will have white and the highest half black. In general, on board b, where b is in the range 1 ... B, player m shall have white against player N-m+1.

Example 10 players:

```
Rd 1:    1 - 10   2 - 9   3 - 8   4 - 7   5 - 6
```

All following rounds are built from the previous rounds with the algorithm:

Step 1:
Sort the players 1 … N-1 according to their pair number in the previous round and then white before black. On the first board player N shall meet the last player in sorted list. Remove this player from the list and pair him against player N. Player N shall alternate color. This other player will be white if his pairing number is in the range 1 … B, and black otherwise.

Example 10 players:

```
Rd 1:    1 - 10   2 - 9   3 - 8   4 - 7   5 - 6
List:    1 2 9 3 8 4 7 5 6
Board 1:    10 - 6
Remaining:    1 2 9 3 8 4 7 5
```

Step 2:
From the end of the list, make a pair with white as the penultimate player in the list, and black as the last player in the list and remove these players from the list. Repeat this until the list is empty.

Example 10 players:

```
Remaining:    1 2 9 3 8 4 7 5
Board 1-2:    10 - 6   7 - 5
Remaining:    1 2 9 3 8 4
Board 1-3:    10 - 6   7 - 5   8 - 4
Remaining:    1 2 9 3
Board 1-4:    10 - 6   7 - 5   8 - 4   9 - 3
Remaining:    1 2
Board 1-5:    10 - 6   7 - 5   8 - 4   9 - 3   1 - 2
```

Repeat from step 1 until all rounds are paired.

After 9 rounds the complete paring is:

```
Rd 1:    1 - 10   2 - 9   3 - 8   4 - 7   5 - 6
Rd 2:    10 - 6   7 - 5   8 - 4   9 - 3   1 - 2
Rd 3:    2 - 10   3 - 1   4 - 9   5 - 8   6 - 7
Rd 4:    10 - 7   8 - 6   9 - 5   1 - 4   2 - 3
Rd 5:    3 - 10   4 - 2   5 - 1   6 - 9   7 - 8
Rd 6:    10 - 8   9 - 7   1 - 6   2 - 5   3 - 4
Rd 7:    4 - 10   5 - 3   6 - 2   7 - 1   8 - 9
Rd 8:    10 - 9   1 - 8   2 - 7   3 - 6   4 - 5
Rd 9:    5 - 10   6 - 4   7 - 3   8 - 2   9 - 1
```

Code in python

```
"""
Created on Sun Nov 12 15:50:51 2023
@author: Otto Milvang, sjakk@milvang.no
bergertables is the core of the round robin, Se FIDE handbook
C: General Rules and Technical Recommendations for Tournaments
05. General Regulations for Competitions / General Regulations for
Competitions. Annex 1: Details of Berger Table

n is he number of players, odd nubers are lifted to the first even number
returns a 3-level dict
  1-dim - 1 .. n-1 is [round_number]
  2-dim - 1 .. n/2 is [pair_number]
  3-dim - {'white', 'black'} - players numbered 1 .. n
rr = bergertables(n)
In round 6, board 2:
 white = rr['parining'][6][2]['white']
 black = rr['parining'][6][2]['black']
"""
def bergertablesGeneric(nplayers):
    nplayers = nplayers  + (nplayers % 2)
    pairs = nplayers//2
    bergertable = { 'players': nplayers, 'parining': {} }
    pairing =  {}
    for board in range(1, pairs + 1):
        pairing[board] = {'white': board, 'black': nplayers - board + 1}
    bergertable['parining'][1] = pairing
    nplayeriswhite = False
    for rnd in range(2, nplayers):
        playerlist = []
        for board in range(1, pairs+1):
            for color in ['white', 'black']:
                if pairing[board][color] < nplayers:
                    playerlist.append(pairing[board][color])
        pairing = {}
        firstboard = playerlist.pop()
        if nplayeriswhite:
            pairing[1] = {'white': firstboard, 'black': nplayers}
        else:
            pairing[1] = {'white': nplayers, 'black': firstboard}
        nplayeriswhite = not nplayeriswhite


        for board in range(2, pairs+1):
            black = playerlist.pop()
            white = playerlist.pop()
            pairing[board] = {'white': white, 'black': black}
        bergertable['parining'][rnd] = pairing
    return bergertable
```

See: https://github.com/OttoMilvang/TieBreakServer/blob/main/berger.py