Testing and enumeration of football sequences

Loránd Lucz, Antal Iványi

Department of Computer Algebra
Eötvös Loránd University
Budapest, Hungary
email: lorand.lucz@gmail.com

11/02/2012

Research problem (Frank, A. [1])

Decide if a sequence of n integers can be the final score of a football tournament of n teams. The winner of a game gets 3 points, the loser no point, while both teams get 1 point for a draw.

Purpose

We would like to give polynomial algorithm to test football sequences and enumerate them.

Used method

Filter potential football sequences with fast linear time approximate algorithms and check only the remaining ones with the slow precise test.

1. Filtering tests
    1. Minimal test
    2. Complete test
    3. Draw-loss test
2. Sport matrices
3. Draw sequences
4. Imbalance sequences

Definition

Q is a **regular** sequence, if $Q = (q_1 \leq \cdots \leq q_n)$ and $0 \leq q_i \leq n-1$.

Definition

Let Q regular sequence is *(l,u,m)*-**bounded** for given l, u and m integer values if $m \geq 1$ and $l \leq q_1 \leq \cdots \leq q_n \leq u$.

Theorem

It is easy to show that the number of *(l,u,m)*-bounded regular sequences is

$$R(l,u,m) = \binom{u-l+m}{m}.$$

Definition

Q is a **minimal tested** sequence if it contains at most one zero.

Theorem

If a Q regular sequence is a football sequence then

$$q_2 > 0.$$

Complexity

For any sequence, minimal test needs exactly 1 operation to check if it is a minimal tested sequence or not.

Theorem (Iványi, A. [3])

Q is a **complete** *(2,3,n)* sequence if and only if

$$2\sum_{i=1}^{k}\binom{i}{2} \le \sum_{i=1}^{k} q_i \le \binom{n}{2} - L_k - (n-k)q_k \quad (k = 1, 2, \ldots, n),$$

where $L_0 = 0$, $L_k = \max\left(L_{k-1}, 3\binom{n}{2} - \sum_{i=1}^{k} q_i\right)$.

Complexity

In best case the test needs exactly 7 operations to reject a non (2,3,n)-complete sequence. In worst case the test needs approximately 16n operations.

Theorem (Iványi, A., Pirzada, S. [6])

Q (2,3,n)-complete sequence is **draw-loss** satisfying if

$$\sum_{i=1}^{k} q_i + (n-k)q_k \leq 3\binom{n}{2} - L'_k \quad (k = 1, 2, \ldots, n),$$

where $L'_0 = 0$, $L'_k = \max\left(L'_{k-1}, 3\binom{n}{2} - \sum_{i=1}^{k} q_i, \left\lceil \frac{\sum_{i=1}^{k}(q_i) - 3q_i/3}{2} \right\rceil\right)$.

Complexity

In best case the test needs exactly 18 operations to reject a non-complete sequence. In worst case the test needs approximately $18n$ operations.

Number of accepted sequences

n	$R(0,3(n-1),n)$	$M(2,3,n)$	$C(2,3,n)$	$DL(2,3,n)$
1	1	1	1	1
2	10	9	4	3
3	84	77	27	19
4	715	660	208	149
5	6 188	5 733	1 709	1 274
6	54 264	50 388	14 513	11 227
7	480 700	447 051	125 658	100 201
8	4 292 145	3 996 135	1 102 081	900 548
9	38 567 100	35 937 525	9 756 399	8 131 268
10	348 330 136	324 794 316	86 989 413	73 680 255
11	3 159 461 968	2 947 546 836	780 019 710	669 601 006
12	28 760 021 745	26 842 686 962	7 026 788 895	6 100 624 883
13	262 596 783 764	245 179 650 147	63 546 151 172	55 704 858 702

Figure 1: Number of regular, minimal-tested, complete, draw-loss sequences.

Ratio of accepted and regular sequences

n	$\frac{M(2,3,n)}{R(0,3(n-1),n)}$	$\frac{C(2,3,n)}{R(0,3(n-1),n)}$	$\frac{DL(2,3,n)}{R(0,3(n-1),n)}$
1	1,0000000000	1,0000000000	1,0000000000
2	0,9000000000	0,4000000000	0,3000000000
3	0,9166666667	0,3214285714	0,2261904762
4	0,9230769231	0,2909090909	0,2083916084
5	0,9264705882	0,2761797027	0,2058823529
6	0,9285714286	0,2674517175	0,2068959163
7	0,9300000000	0,2614062825	0,2084480965
8	0,9310344828	0,2567669545	0,2098130422
9	0,9318181818	0,2529720669	0,2108343121
10	0,9324324324	0,2497326645	0,2115242047
11	0,9329268293	0,2468837156	0,2119351373
12	0,9333333333	0,2443248812	0,2121217062
13	0,9336734694	0,2419913537	0,2121307729

Figure 2: Ratio of minimal and regular, complete and regular, draw-loss and regular sequences.

Increase of number of sequences

n	$\frac{R(0,3(n-1),n)}{R(0,3(n-2),n-1)}$	$\frac{M(2,3,n)}{M(2,3,n-1)}$	$\frac{C(2,3,n)}{C(2,3,n-1)}$	$\frac{DL(2,3,n)}{DL(2,3,n-1)}$
2	10,0000000000	9,0000000000	4,0000000000	3,0000000000
3	8,4000000000	8,5555555556	6,7500000000	6,3333333333
4	8,5119047619	8,5714285714	7,7037037037	7,8421052632
5	8,6545454545	8,6863636364	8,2163461538	8,5503355705
6	8,7692307692	8,7891156463	8,4921006437	8,8124018838
7	8,8585434174	8,8721719457	8,6583063460	8,9250022268
8	8,9289473684	8,9388794567	8,7704801923	8,9874152953
9	8,9855072464	8,9930708047	8,8527059263	9,0292444156
10	9,0317948718	9,0377485929	8,9161393461	9,0613487343
11	9,0703090013	9,0751182850	8,9668349642	9,0879300838
12	9,1028225806	9,1067889521	9,0084760743	9,1108358983
13	9,1306184012	9,1339458860	9,0434125917	9,1310086705

Figure 3: Ratio of i- and $i+1$-regular, minimal, complex and draw-loss sequences.

Complexity of filtering tests

n	$\frac{O_{M(2,3,n)}}{R(0,3(n-1),n)}$	$\frac{O_{C(2,3,n)}}{M(2,3,n)}$	$\frac{O_{C(2,3,n)}}{M(2,3,n)}/n$	$\frac{O_{DL(2,3,n)}}{C(2,3,n)}$	$\frac{O_{DL(2,3,n)}}{C(2,3,n)}/n$
1	1	0,0000	0,0000	0,0000	0,0000
2	1	20,4444	10,2222	25,0000	12,5000
3	1	29,3636	9,7879	36,5556	12,1852
4	1	36,1439	9,0360	47,7308	11,9327
5	1	44,0502	8,8100	59,6870	11,9374
6	1	51,3372	8,5562	71,7324	11,9554
7	1	59,1292	8,4470	83,7667	11,9667
8	1	66,7114	8,3389	95,8061	11,9758
9	1	74,4884	8,2765	107,8426	11,9825
10	1	82,2093	8,2209	119,8747	11,9875

Figure 4: Ratio of operations per sequence and operations per element using minimal, complete and draw-loss tests.

Method to generate sport matrix

- Deal obvious draws: For every q_i, $mod(q_i, 3) = d_i'$ is the number of obvious draws for the i^{th} team.
- Deal non-obvious draws:
  - The number of matches in the tournament is $m = \binom{n}{2}$.
  - If $\sum_{i=1}^{n} v_i + \frac{\sum_{i=1}^{n} d_i}{2} < m$, so there are not enough matches in the matrix, then we add some more with a recursive algorithm. We can give 3 draws to a team at a time, but then we have to take one victory.
- From victories and draws we can calculate the number of losses as $l_i = (n-1) - (v_i + d_i)$.

Example

For the $Q = (1, 1, 6)$ the sport matrix is obvious and it is

V	D	L
0	1	1
0	1	1
2	0	0

Necessary conditions for sport matrices of football tournaments

- Every v_i, d_i and l_i ($i = 1, \ldots, n$) element of a sport matrix is $0 \le q_i \le n - 1$.
- $q_i = 3v_i + d_i$ ($i = 1, \ldots, n$)
- The D (draws) column has to be a *graphical* sequence. Can be checked in linear time with:
    - Erdős-Gallai linear jumping algorithm [4],
    - Havel-Hakimi linear algorithm [2].
- The $(v_1 - l_1, \ldots, v_n - l_n)$ sequence have to be an *imbalance* sequence.

Erdős-Gallai linear algorithm

Theorem (Iványi, A., Lucz, L., Móri, F. T., Sótér, P. [4])

If $n \geq 1$, then the $D = (d_1, \ldots, d_n)$ sequence is graphical if and only if

$$H_n \ \textit{is even}$$

and

$$H_i > i(y_i - 1) + H_n - H_{y_i} \ \ (i = 1, \ldots, n-1),$$

where

$$y_i = \max(i, w_i), \ (i = 1, \ldots, n-1), \ H_k = \sum_{i=1}^{k} d_i.$$

Number of graphical sequences, complexity of the algorithms

n	R	G	O_{EG}/R	$O_{EG}/R/n$	O_{HH}/R	$O_{HH}/R/n$
1	1	1	1,00	1,00	1,00	1,00
2	3	2	4,00	2,00	4,33	2,17
3	10	4	9,40	3,13	7,90	2,63
4	35	11	14,23	3,56	11,54	2,89
5	126	31	18,63	3,73	14,71	2,94
6	462	102	22,61	3,77	18,41	3,07
7	1 716	342	26,43	3,78	21,89	3,13
8	6 435	1 213	30,23	3,78	25,47	3,18
9	24 310	4 361	34,08	3,79	28,88	3,21
10	92 378	16 016	38,00	3,80	32,26	3,23
11	352 716	59 348	42,01	3,82	35,53	3,23
12	1 352 078	222 117	46,11	3,84	38,74	3,23
13	5 200 300	836 315	50,31	3,87	41,88	3,22
14	20 058 300	3 166 852	54,60	3,90	44,97	3,21
15	77 558 760	12 042 620	58,99	3,93	48,00	3,20
16	300 540 195	45 967 479	63,47	3,97	50,99	3,19
17	1 166 803 110	176 005 709	68,05	4,00	53,93	3,17

Figure 5: Number of regular and graphical sequences and complexity of checking algorithms.

Definition

$I = V - L$ is an **imbalance sequence** if the elements of V and L can be paired without pairing any v_i and l_j element when $i = j$.

Necessary condition

If I is an imbalance sequence then $\sum_{j=1}^{n} i_j = 0$.

Theorem (Iványi [5])

A nonincreasing sequence of integers $S = (s_1, \ldots, s_n)$ is an imbalance sequence of a football tournament if and only if s_i $(i = 1, ..., n)$ is a multiple of 3 and $S' = (s_1/3, \ldots, s_n/3)$ is the imbalance sequence of a complete (0, 1, n)-tournament.

Reconstruction of football sequences

Our last step is to reconstruct the result matrix from the sport matrix. The result matrix contains result of every match. Checking of sport matrices can be done in only by backtracking algorithms. We can check draw and imbalance sequences fast, but unfortunately it is not enough to reconstruct the results.

New filtering tests

New linear time tests to decrease average operations needed per sequence by filtering out more non-football sequences.

Enumerate football sequences

Network based distributed software solution to generate and check all $(2, 3, n)$-complete sequences and count the number of football sequences.

Hivatkozások I

[1] Frank, A.
Connections in Combinatorial Approximation.
Oxford University Press, Oxford, 2011.

[2] Iványi, A.
Degree sequences of multigraphs.
Ann. Univ. Sci. Budap. Rolando Eőtvős, Sect. Comput. (submitted).

[3] Iványi, A.
Reconstruction of complete interval tournaments.
Acta Univ. Sapientiae, Inform. **1**, 1, (2009), 71–88.

[4] Iványi, A.
Imbalance sequences of football tournaments.
Acta Univ. Sapientiae, Inform. (submitted).

Hivatkozások II

[5] Iványi, A., Lucz, L., Móri, F. T., Sótér, P.
On Erdős-Gallai and Havel-Hakimi algorithms.
Acta Univ. Sapientiae, Inform. **3**, 2, (2011), 230–268.

[6] Iványi, A., Pirzada, S.
Comparison Based Ranking in *Algorithms of Informatics, Vol. 3.* 1209–1258.
AnTonCom, Budapest, 2011.

[7] Sloane, N. J. A.
Number of graphical partitions.
The On-line Encyclopedia of Integer Sequences. https://oeis.org, 2012