Directed graphs with prescribed score sequences

Antal Iványi*

Department of Computer Algebra
Eötvös Loránd University
1117 Budapest, Pázmány Péter sétány 1/C,
Hungary
tony@compalg.inf.elte.hu

Abstract: Let $a,\ b\ (b \geq a)$ and $n\ (n \geq 2)$ be nonnegative integers and let $\mathcal{T}(a,b,n)$ be the set of such generalized tournaments, in which every pair of distinct players is connected at most with b, and at least with a arcs. In [14] we gave a necessary and sufficient condition to decide whether a given sequence of nonnegative integers $D = (d_1, d_2, \ldots, d_n)$ can be realized as the out-degree sequence of a $T \in \mathcal{T}(a,b,n)$. In this talk we show that for any sequence of nonnegative integers D there exist f and g such that some element $T \in \mathcal{T}(g,f,n)$ has D as its out-degree sequence, and for any (a,b,n)-tournament T' with the same out-degree sequence D hold $a \leq g$ and $b \geq f$. We propose a $\Theta(n)$ algorithm to determine f and g and an $O(d_n n^2)$ algorithm to construct a corresponding tournament T.

We also study the existence, precise and approximate construction of generalized tournaments with prescribed in-degree and out-degree sequences.

Keywords: tournament, out-degree sequence, score sequence

1 Introduction

Let $a,\ b\ (b \geq a)$ and $n\ (n \geq 2)$ be nonnegative integers and let $\mathcal{T}(a,b,n)$ be the set of such generalized tournaments (tournament graphs), in which every pair of distinct players is connected at most with b, and at least with a arcs. The elements of $\mathcal{T}(a,b,n)$ are called *(a,b,n)-tournaments*. The vector $D = (d_1, d_2, \ldots, d_n)$ of the out-degrees of $T \in \mathcal{T}(a,b,n)$ is called *the score vector* of T. If the elements of D are in nondecreasing order, then D is called *the score sequence* of T.

An arbitrary vector $D = (d_1, d_2, \ldots, d_n)$ of nonnegative integers is called *graphical vector,* iff there exists a loopless multigraph whose degree vector is D, and D is called *digraphical vector* (or *score vector*) iff there exists a loopless directed multigraph whose out-degree vector is D.

A nondecreasingly ordered graphical vector is called *graphical sequence*, and a nondecreasingly ordered digraphical vector is called *digraphical sequence* (or *score sequence*).

The number of arcs of T going from player P_i to player P_j is denoted by m_{ij} $(1 \leq i, j \leq n)$, and the matrix $\mathcal{M} = [1.\ .n, 1.\ .n]$ is called *point matrix* or *tournament matrix* of T.

*The European Union and the European Social Fund have provided financial support to the project under the grant agreement no. TÁMOP 4.2.1/B-09/1/KMR-2010-0003

In the last sixty years many efforts were devoted to the study of both types of vectors, resp. sequences. Even in the last three years many authors investigated the conditions, when D is graphical (e.g. [22]) or digraphical (e.g. [2, 14, 15, 16, 20]).

In this paper we deal only with directed graphs and usually follow the terminology used by K. B. Reid [24]. If in the given context a, b and n are fixed or non important, then we speak simply on *tournaments* instead of generalized or (a, b, n)-tournaments.

We consider the loopless directed multigraphs as generalized tournaments, in which the number of arcs from vertex/player P_i to vertex/player P_j is denoted by m_{ij}, where m_{ij} means the number of points won by player P_i in the match with player P_j.

The first question: how one can characterise the set of the score sequences of the (a, b, n)-tournaments. The answer is given in Section 2.

If T is an (a, b, n)-tournament with point matrix $\mathcal{M} = [1..n, 1..n]$, then let $E(T)$, $F(T)$ and $G(T)$ be defined as follows: $E(T) = \max_{1 \le i,j \le n} m_{ij}$, $F(T) = \max_{1 \le i < j \le n}(m_{ij} + m_{ji})$, and $G(T) = \min_{1 \le i < j \le n}(m_{ij} + m_{ji})$. Let $\Delta(D)$ denote the set of all tournaments having D as out-degree sequence, and let $e(D)$, $f(D)$ and $g(D)$ be defined as follows: $e(D) = \{\min\ E(T) \mid T \in \Delta(D)\}$, $f(D) = \{\min\ F(T) \mid T \in \Delta(D)\}$, and $g(D) = \{\max\ G(T) \mid T \in \Delta(D)\}$. In the sequel we use the short notations E, F, G, e, f, g, and Δ.

Hulett et al. [13] investigated the construction problem of a minimal size graph having a prescribed degree set. In a similar way we follow a mini-max approach formulating the following questions: given a sequence D of nonnegative integers,

- How to compute e and how to construct a tournament $T \in \Delta$ characterised by e? In Section 3 a formula to compute e, and an algorithm to construct a corresponding tournament are presented.
- How to compute f and g? In Section 4 an algorithm to compute f and g is described.
- How to construct a tournament $T \in \Delta$ characterised by f and g? In Section 5 an algorithm to construct a corresponding tournament is presented and analysed.

2 Existence of a tournament with arbitrary degree sequence

Since the numbers of points m_{ij} are not limited, it is easy to construct a $(0, d_n, n)$-tournament for any D.

Lemma 1 *If $n \ge 2$, then for any vector of nonnegative integers $D = (d_1, d_2, \ldots, d_n)$ there exists a loopless directed multigraph T with out-degree vector D so, that $E \le d_n$.*

Proof: Let $m_{n1} = d_n$ and $m_{i,i+1} = d_i$ for $i = 1, 2, \ldots, n-1$, and let the remaining m_{ij} values be equal to zero.

Using weighted graphs it would be easy to extend the definition of the (a, b, n)-tournaments to allow *arbitrary real values* of a, b, and D. Naive-Construct works without changes also for input consisting of real numbers.

In the following the algorithms are defined using the pseudocode described in the textbook due to Cormen, Leiserson, Rivest, and Stein [4].

2.1 Definition of a naive reconstructing algorithm

Sorting of the elements of D is not necessary.

Input. n: the number of players ($n \geq 2$);
$D = (d_1, d_2, \ldots, d_n)$: arbitrary sequence of nonnegative integer numbers.

Output. $\mathcal{M} = [1..n, 1..n]$: the point matrix of the reconstructed tournament.

The pseudocode of algorithm Naive-Construct can be found in [15]. Its running time is $\Theta(n^2)$ in worst case (in best case too). Since the point matrix $\mathcal{M}$ has n^2 elements, this algorithm is asymptotically optimal.

3 Computation of e

This is also an easy question. From here we suppose that D is a nondecreasing sequence of nonnegative integers, that is $0 \leq d_1 \leq d_2 \leq \ldots \leq d_n$. Let $h = \lceil d_n/(n-1) \rceil$.

Since $\Delta(D)$ is a finite set for any finite score vector D, $e(D) = \min\{E(T) | T \in \Delta(D)\}$ exists.

Lemma 2 *If $n \geq 2$, then for any sequence $D = (d_1, d_2, \ldots, d_n)$ there exists a $(0, b, n)$-tournament T such that*

$$E \leq h \quad \text{and} \quad b \leq 2h, \tag{1}$$

and h is the smallest upper bound for e, and $2h$ is the smallest possible upper bound for b.

Proof: If all players gather their points in a uniform as possible manner, that is

$$\max_{1 \leq i,j \leq n, i \neq j} m_{ij} - \min_{1 \leq i,j \leq n,\ i \neq j} m_{ij} \leq 1 \quad \text{for } i = 1,\ 2,\ \ldots,\ n, \tag{2}$$

then we get $E \leq h$, that is the bound is valid. Since player P_n has to gather d_n points, the pigeonhole principle implies $E \geq h$, that is the bound is not improvable. $E \leq h$ implies $\max_{1 \leq i < j \leq n} m_{ij} + m_{ji} \leq 2h$. The score sequence $D = (d_1, d_2, \ldots, d_n) = (2n(n-1), 2n(n-1), \ldots, 2n(n-1))$ shows, that the upper bound $b \leq 2h$ is not improvable.

Corollary 3 *If $n \geq 2$, then for any sequence $D = (d_1, d_2, \ldots, d_n)$ holds $e(D) = \lceil d_n/(n-1) \rceil$.*

Proof: According to Lemma 2 $h = \lceil d_n/(n-1) \rceil$ is the smallest upper bound for e.

3.1 Definition of a construction algorithm

The following algorithm constructs a $(0, 2h, n)$-tournament T having $E \leq h$ for any D.

Input. n: the number of players ($n \geq 2$);
$D = (d_1, d_2, \ldots, d_n)$: arbitrary sequence of nonnegative integer numbers.

Output. $\mathcal{M} = [1..n, 1..n]$: the point matrix of the tournament.

The pseudocode of the algorithm Pigeonhole-Construct can be found in [15]. Its running time is $\Theta(n^2)$ in worst case (in best case too). Since the point matrix $\mathcal{M}$ has n^2 elements, this algorithm is asymptotically optimal.

4 Computation of f and g

Let S_i ($i = 1,\ 2,\ \ldots,\ n$) be the sum of the first i elements of D, B_i ($i = 1,\ 2,\ \ldots,\ n$) be the binomial coefficient $n(n-1)/2$. Then the players together can have S_n points only if $fB_n \geq S_n$. Since the score of player P_n is d_n, the pigeonhole principle implies $f \geq \lceil d_n/(n-1) \rceil$.

These observations result the following lower bound for f:

$$f \geq \max\left(\left\lceil \frac{S_n}{B_n} \right\rceil, \left\lceil \frac{d_n}{n-1} \right\rceil\right). \tag{3}$$

If every player gathers his points in a uniform as possible manner then

$$f \leq 2 \left\lceil \frac{d_n}{n-1} \right\rceil. \tag{4}$$

These observations imply a useful characterisation of f.

Lemma 4 *If $n \geq 2$, then for arbitrary sequence $D = (d_1, d_2, \ldots, d_n)$ there exists a (g, f, n)-tournament having D as its out-degree sequence and the following bounds for f and g:*

$$\max\left(\left\lceil \frac{S}{B_n} \right\rceil, \left\lceil \frac{d_n}{n-1} \right\rceil\right) \leq f \leq 2 \left\lceil \frac{d_n}{n-1} \right\rceil, \tag{5}$$

$$0 \leq g \leq f. \tag{6}$$

PROOF: (5) follows from (3) and (4), (6) follows from the definition of f.

It is worth to remark, that if $d_n/(n-1)$ is integer and the scores are identical, then the lower and upper bounds in (5) coincide and so Lemma 4 gives the exact value of F.

In connection with this lemma we consider three examples. If $d_i = d_n = 2c(n-1)$ ($c > 0,\ i = 1,\ 2,\ \ldots,\ n-1$), then $d_n/(n-1) = 2c$ and $S_n/B_n = c$, that is S_n/B_n is twice larger than $d_n/(n-1)$. In the other extremal case, when $d_i = 0$ ($i = 1,\ 2,\ \ldots, n-1$) and $d_n = cn(n-1) > 0$, then $d_n/(n-1) = cn$, $S_n/B_n = 2c$, so $d_n/(n-1)$ is $n/2$ times larger, than S_n/B_n.

If $D = (0, 0, 0, 40, 40, 40)$, then Lemma 4 gives the bounds $8 \leq f \leq 16$. Elementary calculations show that Figure 1 contains the solution with minimal f, where $f = 10$.

Player/Player	P_1	P_2	P_3	P_4	P_5	P_5	Score
P_1	—	0	0	0	0	0	0
P_2	0	—	0	0	0	0	0
P_3	0	0	—	0	0	0	0
P_4	10	10	10	—	5	5	40
P_5	10	10	10	5	—	5	40
P_6	10	10	10	5	5	—	40

Figure 1: Point matrix of a $(0, 10, 6)$-tournament with $f = 10$ for $D = (0, 0, 0, 40, 40, 40)$.

In [14] we proved the following assertion.

Theorem 5 *For $n \geq 2$ a nondecreasing sequence $D = (d_1, d_2, \ldots, d_n)$ of nonnegative integers is the score sequence of some (a, b, n)-tournament if and only if*

$$aB_k \leq \sum_{i=1}^{k} d_i \leq bB_n - L_k - (n-k)d_k \quad (1 \leq k \leq n), \tag{7}$$

where

$$L_0 = 0, \text{ and } L_k = \max\left(L_{k-1},\ bB_k - \sum_{i=1}^{k} d_i\right) \quad (1 \leq k \leq n). \tag{8}$$

The theorem proved by Moon [23], and later by Kemnitz and Dolff [18] for (a, a, n)-tournaments is the special case $a = b$ of Theorem 5. The theorems of Landau [21] and Avery [1] are the special cases $a = b = 1$ of Theorem 5.

4.1 Definition of a testing algorithm

The following algorithm Interval-Test decides whether a given D is a score sequence of an (a, b, n)-tournament or not. This algorithm is based on Theorem 5 and returns $W =$ True if D is a score sequence, and returns $W =$ False otherwise.

Input. a: minimal number of points divided after each match;
b: maximal number of points divided after each match.

Output. W: logical variable ($W =$ True shows that D is an (a, b, n)-tournament).

Local working variables. i: cycle variable;
$L = (L_0, L_1, \ldots, L_n)$: the sequence of the values of the loss function.

Global working variables. n: the number of players ($n \geq 2$);
$D = (d_1, d_2, \ldots, d_n)$: a nondecreasing sequence of nonnegative integers;
$B = (B_0, B_1, \ldots, B_n)$: the sequence of the binomial coefficients;
$S = (S_0, S_1, \ldots, S_n)$: the sequence of the sums of the i smallest scores.

The pseudocode of the algorithm Interval-Test can be found in [15].

In worst case Interval-Test runs in $\Theta(n)$ time even in the general case $0 < a < b$ (n the best case the running time of Interval-Test is $\Theta(n)$). It is worth to mention, that the often referenced Havel-Hakimi algorithm [10, 11, 12] even in the special case $a = b = 1$ decides in $\Theta(n^2)$ time whether a sequence D is digraphical or not.

4.2 Definition of an algorithm computing f and g

The following algorithm is based on the bounds of f and g given by Lemma 4 and the logarithmic search algorithm.

Input. No special input (global working variables serve as input).

Output. b: f (the minimal F);
a: g (the maximal G).

The pseudocode of algorithm MinF-MaxG can be found in [15].

Lemma 6 *Algorithm* MinG-MaxG *computes the values f and g for arbitrary sequence $D = (d_1, d_2, \ldots, d_n)$ in $O(n \log(d_n/(n))$ time.*

PROOF: According to Lemma 4 F is an element of the interval $[\lceil d_n/(n-1)\rceil, \lceil 2d_n/(n-1)\rceil]$ and g is an element of the interval $[0, f]$. Taking into account the running time of the logarithmic search we get that $O(\log(d_n/n))$ calls of INTERVAL-TEST is sufficient, so the $O(n)$ run time of INTERVAL-TEST implies the required running time of MINF-MAXG.

4.3 Computing of f and g in linear time

Analysing Theorem 5 and the work of algorithm MINF-MAXG one can observe that the maximal value of G and the minimal value of F can be computed independently by LINEAR-MINF-MAXG.

Input. No special input (global working variables serve as input).

Output. b: f (the minimal F).

a: g (the maximal G).

The pseudocode of algorithm LINEAR-MINF-MAXG can be found in [15].

Lemma 7 *Algorithm* LINEAR-MING-MAXG *computes the values f and g for arbitrary sequence $D = (d_1, d_2, \ldots, d_n)$ in $\Theta(n)$ time.*

PROOF: Lines 01–03, 07, and 18 require only constant time, lines 04–06, 09–12, and 13–17 require $\Theta(n)$ time, so the total running time is $\Theta(n)$.

5 Tournament with f and g

The reconstruction algorithm SCORE-SLICING2 is based on balancing between additional points (they are similar to „excess", introduced by Brauer et al. [3]) and missing points introduced in [14]. The greediness of the algorithm Havel–Hakimi [10, 12] also characterises this algorithm.

This algorithm is an extended version of the algorithm SCORE-SLICING proposed in [14].

5.1 Definition of the minimax reconstruction algorithm

The work of the slicing program is managed by the following program MINI-MAX.

Input. No special input (global working variables serve as input).

Output. $\mathcal{M} = [1 \,..\, n, 1 \,..\, n]$: the point matrix of the reconstructed tournament.

The pseudocode of algorithm MINI-MAX can be found in [15].

5.2 Definition of the score slicing algorithm

The key part of the reconstruction is the algorithm SCORE-SLICING2 [14].

During the reconstruction process we have to take into account the following bounds:

$$a \le m_{i,j} + m_{j,i} \le b \quad (1 \le i < j \le n); \tag{9}$$

$$\text{modified scores have to satisfy (7);} \tag{10}$$

$$m_{i,j} \le p_i \ (1 \le i,\ j \le n, i \ne j); \tag{11}$$

$$\text{the monotonicity } p_1 \le p_2 \le \ldots \le p_k \text{ has to be saved } \ (1 \le k \le n) \tag{12}$$

$$m_{ii} = 0 \quad (1 \le i \le n). \tag{13}$$

Input. k: the number of the actually investigated players ($k > 2$);
$\mathbf{p}_k = (p_0, p_1, p_2, \ldots, p_k)$ ($k = 3,\ 4,\ \cdots,\ n$): prefix of the provisional score sequence p;
$\mathcal{M}[1\ ..\ n, 1\ ..\ n]$: matrix of provisional points.

Output. $\mathcal{M}[1\ ..\ n, 1\ ..\ n]$: matrix of provisional points;
$\mathbf{p}_k = (p_0, p_1, p_2, \ldots, p_k)$ ($k = 2,\ 3,\ 4,\ \cdots,\ n-1$): prefix of the provisional score sequence p.

This algorithm is a corrected version of algorithm ScoreSlicing whose pseudocode can be found in [15].

Let's consider an example. Figure 2 shows the point table of a $(2, 10, 6)$-tournament T.

Player/Player	P_1	P_2	P_3	P_4	P_5	P_6	Score
P_1	—	1	5	1	1	1	9
P_2	1	—	4	2	0	2	9
P_3	3	3	—	5	4	4	19
P_4	8	2	5	—	2	3	20
P_5	9	9	5	7	—	2	32
P_6	8	7	5	6	8	—	34

Figure 2: The point table of a $(2, 10, 6)$-tournament T.

The score sequence of T is $D = (9,9,19,20,32,34)$. In [14] the algorithm Score-Slicing2 resulted the point table represented in Figure 3.

Player/Player	P_1	P_2	P_3	P_4	P_5	P_6	Score
P_1	—	1	1	6	1	0	9
P_2	1	—	1	6	1	0	9
P_3	1	1	—	6	8	3	19
P_4	3	3	3	—	8	3	20
P_5	9	9	2	2	—	10	32
P_6	10	10	7	7	0	—	34

Figure 3: The point table of T reconstructed by Score-Slicing2.

The algorithm Mini-Max returns the point matrix represented in Figure 4.

The comparison of Figures 3 and 4 shows a large difference between the simple reconstruction of Score-Slicing2 and the minimax reconstruction of Mini-Max.

Comparing the construction algorithm Mini-Max with the algorithms of S. V. Gervacio [8, 9] or with the algorithm A. Kemnitz and S. Dolff [18] we can say that our algorithm not only construct a corresponding tournament but constructs in the given sense optimal tournament.

6 Main result

Theorem 8 *If $n \ge 2$ is a positive integer and $D = (d_1, d_2, \ldots, d_n)$ is a nondecreasing sequence of nonnegative integers, then there exist positive integers f and g, and a (g, f, n)-tournament T*

Player/Player	P_1	P_2	P_3	P_4	P_5	P_6	Score
P_1	—	4	4	1	0	0	9
P_2	4	—	4	1	0	0	9
P_3	4	4	—	7	4	0	19
P_4	7	7	1	—	5	0	20
P_5	8	8	4	3	—	9	32
P_6	9	9	8	8	0	—	34

Figure 4: The point table of T reconstructed by Mini-Max.

with point matrix $\mathcal{M}$ such, that

$$f = \min(m_{ij} + m_{ji}) \leq b, \tag{14}$$

$$g = \max(m_{ij} + m_{ji}) \geq a \tag{15}$$

for any (a, b, n)-tournament, and algorithm Linear-MinF-MaxG *computes f and g in $\Theta(n)$ time, and algorithm* Mini-Max *generates a suitable T in $O(d_n n^2)$ time.*

Proof: The correctness of the algorithms Score-Slicing2, Min$F$-Max$G$ implies the correctness of Mini-Max.

Lines 1–46 of Mini-Max require $O(\log(d_n/n))$ uses of MinG-MaxF, and one search needs $O(n)$ steps for the testing, so the computation of f and g can be executed in $O(n \log(d_n/n))$ times.

The reconstruction part (lines 47–55) uses algorithm Score-Slicing2, which runs in $O(bn^3)$ time [14]. Mini-Max calls Score-Slicing2 $n - 2$ times with $f \leq 2\lceil d_n/n \rceil$, so $n^3 d_n/n = d_n n^2$ finishes the proof.

The property of the tournament reconstruction problem that the extremal values of f and g can be determined independently and so there exists a tournament T having both extremal features is called linking property. This concept was introduced by Ford and Fulkerson in 1962 [6] and later extended by A. Frank in [7].

7 Further problems

The vector vector $L = (l_1, l_2, \ldots, l_n)$ of the in-degrees of $T \in \mathcal{T}(a, b, n)$ is called *the losing vector* of T.

Hakimi proved the following theorem in 1965 [11], whose special case $a = b = 1$ was later reproved by H. Kim, Z. Toroczkai, I. Miklós, P. L. Erdős and L. A. Székely [19].

Theorem 9 *Let $n \geq 2$ be a positive integer, $D = (d_1, d_2, \ldots, d_n)$ and $L = (l_1, l_2, \ldots, l_n)$ two vectors of nonnegative integers, which are ordered are so that $d_i + l_i \leq d_{i+1} + l_{i+1}$ for $i = 1, 2, \ldots, n-1$, and $d_1 + l_1 > 0$. There exists a tournament $T \in \mathcal{T}(a, \infty, n)$ with score vector D and losing vector L if and only if*

$$\sum_{i=1}^{n} d_i = \sum_{i=1}^{n} l_i \tag{16}$$

and

$$\sum_{i=1}^{n}(d_i + l_i) \geq d_n + l_n. \tag{17}$$

Using this theorem one can decide in linear time whether a given pair (D, L) is the score vector-losing vector pair of some $T \in \mathcal{T}(a, \infty, n)$ or not. In the talk we investigate the following two problems too.

a) If D is the score vector and L and is the losing vector of some tournament $T \in \mathcal{T}(a, \infty, n)$ such, that they satisfy all conditions of Theorem 9, how we can construct a balanced as possible tournament $T' \in \mathcal{T}(a, \infty, n)$, having the same score and losing vector.

b) If D is the score vector and L and is the losing vector of some tournament $T \in \mathcal{T}(a, \infty, n)$ such, that they are ordered according to 9, but they do **not** satisfy (16) and (17), then how we can define and construct an approximate solution.

8 Summary

A nondecreasing sequence of nonnegative integers $D = (d_1, d_2, \ldots, d_n)$ is a score sequence of a $(1, 1, 1)$-tournament, iff the sum of the elements of D equals to B_n and the sum of the first i $(i = 1,\ 2,\ \ldots,\ n-1)$ elements of D is at least B_i [21].

D is a score sequence of an (a, b, n)-tournament, iff (7) holds [14]. The decision whether D is digraphical requires only linear time. In this talk we prove that for any D there exists an optimal minimax realization T, that is a tournament having D as its out-degree sequence, and maximal G, and minimal F in the set of all realizations of D.

Acknowledgement. The author thanks András Frank (Eötvös Loránd University) and Péter L. Erdős (Alfréd Rényi Institute of Mathematics of HAS) for the useful consultation.

References

[1] P. Avery, Score sequences of oriented graphs. *J. Graph Theory* (1991) **15** (3) 251–257.

[2] L. B. Beasley, D. E. Brown, K. B. Reid, Extending partial tournaments, *Math. Comput. Modelling* (2009) **50** (1) 187–291.

[3] A. Brauer, I. C. Gentry, K. Shaw, A new proof of a theorem by H. G. Landau on tournament matrices, *J. Comb. Theory* (1968) **5** 289–292.

[4] T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms.* Third edition, MIT Press/McGraw Hill, Cambridge/New York, 2009.

[5] P. Erdős, T. Gallai, Graphs with prescribed degrees of vertices (Hungarian), *Mat. Lapok* (1960) **11** 264–274.

[6] L. R. Ford, D. R. Fulkerson, *Flows in Networks*, Princeton University, Press, Princeton (1960).

[7] A. Frank, On the orientation of graphs, *J. Combin. Theory, Ser. B.* (1980) **3** 251–261.

[8] S. V. Gervacio, Score sequences: Lexicographic enumeration and tournament construction *Discrete Math.* (1988) **72** (1–3) 151–155.

[9] S. V. Gervacio, Construction of tournaments with a given score sequence. *Southeast Asian Bull. Math.* (1993) **17** (2) 151–155.

[10] S. L. Hakimi, On the realizability of a set of integers as degrees of the vertices of a simple graph, *J. SIAM Appl. Math.* (1962) **10** 496–506.

[11] S. L. Hakimi, On the degrees of the vertices of a directed graph, *J. Franklin Inst.* (1965) **279** 290–308.

[12] V. Havel, A remark on the existence of finite graphs (Czech), *Časopis Pěst. Mat.* (1955) **80** 477–480.

[13] H. Hulett, T. G. Will, G. J. Woeginger, Multigraph realizations of degree sequences: Maximization is easy, minimization is hard, *Operations Research Letters* (2008) **36** 594–596.

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

[15] A. Iványi, Reconstruction of interval tournaments 2, *Acta Univ. Sapientiae, Mathematica* (2010) **2** (1) 47–71.

[16] A. Iványi, S. Pirzada, Comparison based ranking, In *Algorithms of Informatics. Volume 3* (ed. by A. Iványi). AnTonCom, Budapest, 2011 (to appear).

[17] S. F. Kapoor, A. D. Polimeni, C. E. Wall, Degree sets for graphs, *Fund. Math.* (1977) **95** 189–194.

[18] A. Kemnitz, S. Dolff, Score sequences of multitournaments, *Congr. Numer.* (1997) **127** 85-195.

[19] H. Kim, Z. Toroczkai, I. Miklós, P. L. Erdős, L. A. Székely, Degree-based graph construction, *J. Physics: Math. Theor. A* (2009) **42** (39) 392001 (10 pp).

[20] D. E. Knuth, *The Art of Computer Programming. Volume 4A. Combinatorial Algorithms*, Addison-Wesley, Princeton (2011).

[21] H. G. Landau, On dominance relations and the structure of animal societies. III., *Bull. Math. Biophys.* (1953) **15** 143–148.

[22] D. Meierling, L. Volkmann, A remark on degree sequences of multigraphs, *Math. Methods Oper. Res.* (2009) **69** (2) 369–374.

[23] J. W. Moon, An extension of Landau's theorem on tournaments, *Pacific J. Math.* (1963) **13** 1343–1345.

[24] K. B. Reid, Tournaments. In *Handbook of Graph Theory* (ed. by J. L. Gross and J. Yellen), CRC Press, Boca Raton, 2004. (2008) **60** (3) 187–191.