Acta Univ. Sapientiae, Mathematica, **2**, 1 (2010) 47–71

Reconstruction of complete interval tournaments. II.

Antal Iványi
Eötvös Loránd University,
Department of Computer Algebra
1117 Budapest, Pázmány P. sétány 1/C.
email: 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 generalised tournaments, in which every pair of distinct players is connected at most with b, and at least with a arcs. In [40] 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 $T \in \mathcal{T}(a,b,n)$. Extending the results of [40] 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.

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 generalised tournaments, 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.

2010 Mathematics Subject Classification: 05C20, 68C25
Key words and phrases: score sequences, tournaments, efficiency of algorithms

An arbitrary vector $D = (d_1, d_2, \dots, d_n)$ 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} $(1 \le i, j \le n)$, and the matrix $\mathcal{M} = [1..n, 1..n]$ is called *point matrix* or *tournament matrix* of T.

In the last sixty years many efforts were devoted to the study of both types of vectors, resp. sequences. E.g. in the papers [8, 16, 18, 19, 20, 21, 26, 30, 32, 34, 36, 45, 68, 84, 85, 88, 90, 98] the graphical sequences, while in the papers [1, 2, 3, 7, 8, 11, 17, 27, 28, 29, 31, 33, 37, 49, 48, 50, 55, 58, 57, 60, 61, 62, 64, 65, 66, 69, 78, 79, 82, 94, 86, 87, 97, 100, 101] the score sequences were discussed.

Even in the last two years many authors investigated the conditions, when D is graphical (e.g. [4, 9, 12, 13, 22, 23, 24, 25, 38, 39, 43, 47, 51, 52, 59, 75, 81, 92, 93, 95, 96, 104]) or digraphical (e.g. [5, 35, 40, 46, 54, 56, 63, 67, 70, 71, 72, 73, 74, 83, 87, 89, 102]).

In this paper we deal only with directed graphs and usually follow the terminology used by K. B. Reid [79, 80]. If in the given context a, b and n are fixed or non important, then we speak simply on *tournaments* instead of generalised or (a, b, n)-tournaments.

We consider the loopless directed multigraphs as generalised 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. Or, with another words, for which sequences D of nonnegative integers does exist an (a, b, n)-tournament whose out-degree sequence is D. 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. [39, 99], Kapoor et al. [44], and Tripathi et al. [91, 92] investigated the construction problem of a minimal size graph having a prescribed degree set [77, 103]. 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.

We describe the proposed algorithms in words, by examples and by the pseudocode used in [14].

Researchers of these problems often mention different applications, e.g. in biology [55], chemistry Hakimi [32], and Kim et al. in networks [47].

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* $n \geq 2$*, 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* $E \leq d_n$.

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. The following algorithm Naive-Construct works without changes also for input consisting of real numbers.

We remark that Ore in 1956 [66] gave the necessary and sufficient conditions of the existence of a tournament with prescribed in-degree and out-degree vectors. Further Ford and Fulkerson [17, Theorem11.1] published in 1962 necessary and sufficient conditions of the existence of a tournament having prescribed lower and upper bounds for the in-degree and out-degree of the vertices. They results also can serve as basis of the existence of a tournament having arbitrary out-degree sequence.

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.

Working variables. i, j: cycle variables.

Naive-Construct(n, D)

```
for i ← 1 to n
     for j ← 1 to n
          do m_ij ← 0
m_n1 ← d_n
for i ← 1 to n − 1
     do m_i,i+1 ← d_i
return M
```

The running time of this algorithm 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*

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

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

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

then we get $E \le h$, that is the bound is valid. Since player P_n has to gather d_n points, the pigeonhole principle [6, 15, 42] implies $E \ge h$, that is the bound is not improvable. $E \le h$ implies $\max_{1 \le i < j \le n} m_{ij} + m_{ji} \le 2h$. 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 $b \le 2h$ is not improvable. ■

Corollary 1 *If* $n \ge 2$*, 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 $E \le h$ for any D.

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

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

Working variables. i, j, l: cycle variables;
k: the number of the "larger parts" in the uniform distribution of the points.

```
Pigeonhole-Construct(n, D)
for i ← 1 to n
    do m_ii ← 0
       k ← d_i − (n − 1)⌊d_i/(n − 1)⌋
    for j ← 1 to k
        do l ← i + j (mod n)
           m_il ← ⌈d_n/(n − 1)⌉
    for j ← k + 1 to n − 1
        do l ← i + j (mod n)
           m_il ← ⌊d_n/(n − 1)⌋
return M
```

The running time of PIGEONHOLE-CONSTRUCT 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 3 *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 3 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.

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 $D = (0, 0, 0, 40, 40, 40)$.

If $D = (0, 0, 0, 40, 40, 40)$, then Lemma 3 gives the bounds $8 \leq f \leq 16$. Elementary calculations show that Figure 1 contains the solution with minimal f, where $f = 10$.

In [40] we proved the following assertion.

Theorem 1 *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

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

The theorem proved by Moon [61], and later by Kemnitz and Dolff [46] for (a, a, n)-tournaments is the special case $a = b$ of Theorem 1. Theorem 3.1.4 of [22] is the special case $a = b = 2$. The theorem of Landau [55] is the special case $a = b = 1$ of Theorem 1.

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 1 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.

```
Interval-Test(a, b)

for i ← 1 to n
    do L_i ← max(L_{i-1}, bB_n − S_i − (n − i)d_i)
       if S_i < aB_i
         then W ← False
              return W
       if S_i > bB_n − L_i − (n − i)d_i
         then W ← False
              return W
return W
```

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 [32, 36] 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 3 and the logarithmic search algorithm described by D. E. Knuth [53, page 410].

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

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

Local working variables. i: cycle variable;
l: lower bound of the interval of the possible values of F;
u: upper bound of the interval of the possible values of F.

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;
W: logical variable (its value is TRUE, when the investigated D is a score sequence).

MINF-MAXG

```
01 B_0 ← S_0 ← L_0 ← 0                         ▷ Initialisation
02 for i ← 1 to n
03     do B_i ← B_{i-1} + i - 1
04        S_i ← S_{i-1} + d_i
05 l ← max(⌈S_n/B_n⌉, ⌈d_n/(n - 1)⌉)
06 u ← 2 ⌈d_n/(n - 1)⌉
07 W ← TRUE                                    ▷ Computation of f
08 INTERVAL-TEST(0, l)
09 if W = TRUE
10    then b ← l
11         go to 21
12 b ← ⌈(l + u)/2⌉
13 INTERVAL-TEST(0, f)
14 if W = TRUE
15    then go to 17
16 l ← b
17 if u = l + 1
18    then b ← u
19         go to 21
20 go to 14
21 l ← 0                                       ▷ Computation of g
22 u ← f
23 INTERVAL-TEST(b, b)
24 if W = TRUE
25    then a ← f
26         go to 37
27 a ← ⌈(l + u)/2⌉
28 INTERVAL-TEST(0, a)
29 if W = TRUE
30    then l ← a
31         go to 33
```

```
32 u ← a
33 if u = l + 1
34    then a ← l
35         go to 37
36 go to 27
37 return a, b
```

MinF-MaxG determines f and g.

Lemma 4 *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 3 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 $[0, f]$. Using Theorem B of [53, page 412] 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 1 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).
Local working variables. i: cycle variable.
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.

Linear-MinF-MaxG

```
01 B_0 ← S_0 ← L_0 ← 0                 ▷ Initialisation
02 for i ← 1 to n
03     do B_i ← B_{i−1} + i − 1
04        S_i ← S_{i−1} + d_i
05 a ← 0
06 b ← min 2 ⌈d_n/(n − 1)⌉
07 for i ← 1 to n                       ▷ Computation of g
```

08   **do** $a_i \leftarrow \lceil 2S_i/(n^2 - n) \rceil$
09   **if** $a_i > a$
10   **then** $a \leftarrow a_i$
11 **for** $i \leftarrow 1$ **to** n $\triangleright$ Computation of f
12   **do** $L_i \leftarrow \max(L_{i-1}, bB_n - S_i - (n - i)d_i)$
13         $b_i \leftarrow (S_i + (n - i)d_i + L_i)/B_i$
14   **if** $b_i < b$
15   **then** $b \leftarrow b_i$
16 **return** a, b

Lemma 5 *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 following reconstruction algorithm Score-Slicing2 is based on balancing between additional points (they are similar to ,,excess”, introduced by Brauer et al. [10]) and missing points introduced in [40]. The greediness of the algorithm Havel–Hakimi [32, 36] also characterises this algorithm.

This algorithm is an extended version of the algorithm Score-Slicing proposed in [40].

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

Local working variables. i, j: cycle variables.

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;
$p = (p_0, p_1, \ldots, p_n)$: provisional score sequence;
$P = (P_0, P_1, \ldots, P_n)$: the partial sums of the provisional scores;
$\mathcal{M}[1 \ldots n, 1 \ldots n]$: matrix of the provisional points.

Mini-Max

```
MinF-MaxG                              ▷ Initialisation
p_0 ← 0
for i ← 1 to n
    do for j ← 1 to i − 1
           do M[i, j] ← b
       for j ← i to n
           do M[i, j] ← 0
    p_i ← d_i
if n ≥ 3                               ▷ Score slicing for n ≥ 3 players
   then for k ← n downto 3
            do Score-Slicing2(k, p_k, M)
if n = 2                               ▷ Score slicing for 2 players
   then m_{1,2} ← p_1
        m_{2,1} ← p_2
return M
```

5.2 Definition of the score slicing algorithm

The key part of the reconstruction is the following algorithm Score-Slicing2 [40].

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

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

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

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

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

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

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.

Local working variables. $A = (A_1, A_2, \ldots, A_n)$: the number of the additional points;

M: missing points (the difference of the number of actual points and the number of maximal possible points of P_k);
d: difference of the maximal decreasable score and the following largest score;
y: minimal number of sliced points per player;
f: frequency of the number of maximal values among the scores p_1, p_2, $\ldots$, p_{k-1};
i, j: cycle variables;
m: maximal amount of sliceable points;
$P = (P_0, P_1, \ldots, P_n)$: the sums of the provisional scores;
x: the maximal index i with $i < k$ and $m_{i,k} < b$.

Global working variables. n: the number of players ($n \geq 2$);
$B = (B_0, B_1, B_2, \ldots, B_n)$: the sequence of the binomial coefficients;
a: minimal number of points divided after each match;
b: maximal number of points divided after each match.

Score-Slicing2$(k, \mathbf{p}_k, \mathcal{M})$

$P_0 \leftarrow 0$ $\triangleright$ Initialisation
for $i \leftarrow 1$ **to** $k-1$
  do $P_i \leftarrow P_{i-1} + p_i$
    $A_i \leftarrow P_i - aB_i$
$M \leftarrow (k-1)b - p_k$
while $M > 0$ **and** $A_{k-1} > 0$ $\triangleright$ There are missing and additional points
  do $x \leftarrow k-1$
    while $r_{x,k} = b$
      do $x \leftarrow x-1$
    $f \leftarrow 1$
    while $p_{x-f+1} = p_{x-f}$
      do $f = f+1$
    $d \leftarrow p_{x-f+1} - p_{x-f}$
    $m \leftarrow \min(b, d, \lceil A_x/f \rceil, \lceil M/f \rceil)$
    for $i \leftarrow f$ **downto** 1
      do $y \leftarrow \min(b - m_{x+1-i,k}, m, M, A_{x+1-i}, p_{x+1-i})$
        $m_{x+1-i,k} \leftarrow m_{x+1-i,k} + y$
        $p_{x+1-i} \leftarrow p_{x+1-i} - y$
        $m_{k,x+1-i} \leftarrow m_{k,x+1-i} - m_{x+1-i,k}$
        $M \leftarrow M - y$
    for $j \leftarrow i$ **downto** 1
      $A_{x+1-i} \leftarrow A_{x+1-i} - y$
while $M > 0$ and $A_{k-1} = 0$ $\triangleright$ No additional points

```
24         do for i ← k − 1 downto 1
25             y min(m_{k,i}, M, m_{k,i+m_{i,k}−a})
26             m_{ki} ← m_{k,i} − y
27             M ← M − y
28 return p_k, 𝓜
```

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 [40] 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 starts with the computation of f. MinF-MaxG called in line 01 begins with initialisation, including provisional setting of the elements of $\mathcal{M}$ so, that $m_{ij} = b$, if $i > j$, and $m_{ij} = 0$ otherwise. Then MinF-MaxG sets the lower bound $l = \max(9, 7) = 9$ of f in line 05 and tests it in line 08 by Interval-Test. The test shows that $l = 9$ is large enough so Mini-Max sets $b = 9$ in line 12 and jumps to line 21 and begins to compute g. Interval-Test called in line 23 shows that $a = 9$ is too large, therefore

MinF-MaxG continues with the test of $a = 5$ in line 27. The result is positive, therefore comes the test of $a = 7$, then the test of $a = 8$. Now $u = l + 1$ in line 33, so $a = 8$ is fixed, and the control returns to line 02 of Mini-Max.

Lines 02–08 contain initialisation, and Mini-Max begins the reconstruction of a $(8, 9, 6)$-tournament in line 9. The basic idea is that Mini-Max successively determines the won and lost points of P_6, P_5, P_4 and P_3 by repeated calls of Score-Slicing2 in line 11, and finally it computes directly the result of the match between P_2 and P_1 in lines 12–14.

At first Mini-Max computes the results of P_6 calling Score-Slicing2 with parameter $k = 6$. The number of additional points of the first five players is $A_5 = 89 - 8 \cdot 10 = 9$ according to line 04, the number of missing points of P_6 is $M = 5 \cdot 9 - 34 = 11$ according to line 05. Then Score-Slicing2 determines the number of maximal numbers among the provisional scores $p_1, p_2, \ldots, p_5$ ($f = 1$ according to lines 10–12) and computes the difference between p_5 and p_4 ($d = 12$ according to line 13). In line 14 we get, that $m = 9$ points are sliceable, and P_5 gets these points in the match with P_6 in line 17, so the number of missing points of P_6 decreases to $M = 11 - 9 = 2$ (line 20) and the number of additional point decreases to $A_5 = 9 - 9 = 0$. Therefore the computation continues in lines 23–28 and m_{64} and m_{63} will be decreased by 1 resulting $m_{64} = 8$ and $m_{63} = 8$ as the seventh line and seventh column of Figure 4 show. The returned score sequence is $\mathbf{p_5} = (9, 9, 19, 20, 23)$.

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.

Second time Mini-Max calls Score-Slicing2 with parameter $k = 5$, and get $A_4 = 9$ and $M = 13$. At first P_4 gets 1 point, then P_3 and P_4 get both 4 points, reducing M to 4 and A_4 to 0. The computation continues in line 23 and results the further decrease of m_{54}, m_{53}, m_{52}, and m_{51} by 1, resulting $m_{54} = 3$, $m_{53} = 4$, $m_{52} = 8$, and $m_{51} = 8$ as the sixth row of Figure 4 shows. The returned score sequence is $\mathbf{p_4} = (9, 9, 15, 15)$

Third time Mini-Max calls Score-Slicing2 with parameter $k = 4$, and get $A_3 = 11$ and $M = 11$. At first P_3 gets 6 points, then P_3 further 1 point, and P_2 and P_1 also both get 1 point, resulting $m_{34} = 7$, $m_{43} = 2$, $m_{42} = 8$, $m_{24} = 1$, $m_{14} = 1$ and $m_{14} = 8$, further $A_3 = 0$ and $M = 2$. The computation continues in lines 23–28 and results a decrease of m_{43} by 1 point resulting $m_{43} = 1$, $m_{42} = 7$, and $m_{41} = 7$, as the fifth row and fifth column of Figure 4 show. The returned score sequence is $\mathbf{p}_3 = (8, 8, 8)$.

Fourth time Mini-Max calls Score-Slicing2 with parameter $k = 3$, and gets $A_2 = 8$ and $M = 10$. At first P_1 and P_2 get 4 points, resulting $m_{13} = 4$, and $m_{23} = 4$, and $M = 2$, and $A_2 = 0$. Then Mini-Max sets in lines 23–26 $m_{31} = 4$ and $m_{32} = 4$. The returned score sequence is $\mathbf{p}_2 = (4, 4)$.

Finally Mini-Max sets $m_{12} = 4$ and $m_{21} = 4$ in lines 14–15 and 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: while in the first case the maximal value of $m_{ij} + m_{ji}$ is 10 and the minimal value is 2, in the second case the maximum equals to 9 and the minimum equals to 8, that is the result is more balanced (the given D does not allow to build a perfectly balanced (k, k, n)-tournament).

5.3 Analysis of the minimax reconstruction algorithm

The main result of this paper is the following assertion.

Theorem 2 *If $n \geq 2$ 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 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, MinF-MaxG 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 [40]. 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 [17] and later extended by A. Frank in [22].

6 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 [55].

D is a score sequence of a (k,k,n)-tournament, iff the sum of the elements of D equals to kB_n, and the sum of the first i elements of D is at least kB_i [46, 60].

D is a score sequence of an (a,b,n)-tournament, iff (7) holds [40].

In all 3 cases the decision whether D is digraphical requires only linear time.

In this paper the results of [40] are extended proving 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.

In a continuation [41] of this paper we construct balanced as possible tournaments in a similar way if not only the out-degree sequence but the in-degree sequence is also given.

Acknowledgement

The author thanks András Frank (Eötvös Loránd University) for valuable advises concerning the application of flow theory, Péter L. Erdős (Alfréd Rényi Institute of Mathematics of HAS) for the consultation and the unknown referee proposing corrections of the manuscript.

The research was supported by the project TÁMOP-4.2.1.B-09/1/KMR–2010-003 of Eötvös Loránd University.

References

[1] P. Acosta, A. Bassa, A. Chaikin, A. Riehl, A. Tingstad, L. Zhao, D. J. Kleitman, On a conjecture of Brualdi and Shen on block transitive tournaments, *J. Graph Theory*, **44** (2003), 215–230.

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

[3] Ch. M. Bang, H. Sharp, Jr., Score vectors of tournaments, *J. Combin. Theory Ser. B*, **26** (1979), 81–84.

[4] M. D. Barrus, M. Kumbhat, S. G. Hartke, Graph classes characterized both by forbidden subgraphs and degree sequences, *J. Graph Theory*, **57** (2008), 131–148.

[5] L. B. Beasley, D. E. Brown, K. B. Reid, Extending partial tournaments, *Math. Comput. Modelling*, **50** (2009), 287–291.

[6] A. Bege, Z. Kása, *Algorithmic combinatorics and number theory* (Hungarian), Presa Universitară Clujeană, 2006.

[7] M. Belica, Segments of score sequences, *Novi Sad J. Math.*, **30** (2000), 11–14.

[8] C. Berge, *Graphs and hypergraphs* (second, revised edition), North-Holland, Amsterdam, 1976.

[9] F. Boesch, F. Harary, Line removal algorithms for graphs and their degree lists, Special issue on large-scale networks and systems, *IEEE Trans. Circuits and Systems*, **CAS-23** (1976), 778–782.

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

[11] A. R. Brualdi, J. Shen, Landau's inequalities for tournament scores and a short proof of a theorem on transitive sub-tournaments, *J. Graph Theory*, **38** (2001), 244–254.

[12] A. R. Brualdi, K. Kiernan, Landau's and Rado's theorems and partial tournaments, *Electron. J. Combin.*, **16** #N2 (6 pp) (2009).

[13] F. Chung, R. Graham, Quasi-random graphs with given degree sequences, *Random Struct. Algorithms*, **32** (2008), 1–19.

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

[15] J. A. Dossey, A. D. Otto, L. E. Spence, Ch. V. Eynden, *Discrete Mathematics* (4. edition), Addison Wesley, Upper Saddle River, 2001.

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

[17] L. R. Ford, D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, 1962.

[18] A. Frank, A. Gyárfás, How to orient the edges of a graph?, In *Combinatorics. Vol. 1* (ed. A. Hajnal and V. T. Sós), North-Holland, Amsterdam-New York, 1978, pp. 353–364.

[19] A. Frank, On the orientation of graphs, *J. Combin. Theory, Ser. B.*, **28** 3 (1980), 251–261.

[20] A. Frank, Orientations of graphs and submodular functions, *Congr. Num.* **113** (1996), 111–142.

[21] A. Frank, T. Király, Z. Király, On the orientation of graphs and hypergraphs, *Discrete Appl. Math.*, **131** (2003), 385–400.

[22] A. Frank, Connections in combinatorial optimization, I. Optimization in graphs (Hungarian), *Mat. Lapok*, **14** (2008), 20–76.

[23] A. Frank, Connections in combinatorial optimization. II. Submodular optimization and polyhedral combinatorics (Hungarian), *Mat. Lapok*, **14** (2008), 14–75.

[24] A. Frank, L. C. Lap, J. Szabó, A note on degree-constrained subgraphs, *Discrete Math.*, **308** (2008), 2647–2648.

[25] A. Frank, Rooted k-connections in digraphs, *Discrete Appl. Math.*, **157** 6 (2009), 1242–1254.

[26] D. R. Fulkerson, Zero-one matrices with zero trace, *Pacific J. Math.*, **10** (1960), 831–836.

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

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

[29] J. Griggs, K. B. Reid, Landau's theorem revisited, *Australas. J. Comb.*, **20** (1999), 19–24.

[30] J. Griggs, D. J. Kleitman, Independence and the Havel–Hakimi residue, *Discrete Math.*, **127** (1994), 209–212.

[31] B. Guiduli, A. Gyárfás, S. Thomassé, P. Weidl, 2-partition-transitive tournaments, *J. Combin. Theory Ser. B*, **72** (1998), 181–196.

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

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

[34] S. L. Hakimi, On the existence of graphs with prescribed degrees and connectivity, *SIAM J. Appl. Math.*, **26** (1974), 154–164.

[35] H. Harborth, A. Kemnitz, Eine Anzahl der Fussballtabelle, *Math. Semester.*, **29** (1982), 258–263.

[36] V. Havel, A remark on the existence of finite graphs (Czech), *Časopis Pěst. Mat.*, **80** (1965), 477–480.

[37] R. Hemasinha, An algorithm to generate tournament score sequences, *Math. Comp. Modelling*, **37** (2003), 377–382.

[38] L. Hu, C. Lai, P. Wang, On potentially $\mathsf{K}_5 - \mathsf{H}$-graphic sequences, *Czechoslovak Math. J.*, **59 (134)** (2009), 173–182.

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

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

[41] A. Iványi, Balanced digraphs with prescribed degree sequences, *Annales Univ. Sci. Budapest., Sectio Computatorica* (*submitted*).

[42] A. Járai (editor): *Introduction to Mathematics with Applications in Informatics* (Third, corrected and extended edition, Hungarian), ELTE Eötvös Kiadó, Budapest, 2009.

[43] H. Jordon, R. McBride, S. Tipnis, The convex hull of degree sequences of signed graphs, *Discrete Math.*, **309** (2009), 5841–5848.

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

[45] G. Katona, G. Korvin, Functions defined on a directed graph, In *Theory of Graphs* (Proc. Colloq., Tihany, 1966). Academic Press, New York, 1968, pp. 209–213.

[46] A. Kemnitz, S. Dolff, Score sequences of multitournaments, *Congr. Numer.*, **127** (1997), 85–95.

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

[48] D. J. Kleitman, D. L. Wang, Algorithms for constructing graphs and digraphs with given valences and factors, *Discrete Math.*, **6** (1973), 79–88.

[49] D. J. Kleitman, D. L. Wang, Decomposition of a graph realizing a degree sequence into disjoint spanning trees, *SIAM J. Appl. Math.*, **30** (1976), 206–221.

[50] D. J. Kleitman, K. J. Winston, Forests and score vectors, *Combinatorica*, **1** (1981), 49–51.

[51] C. J. Klivans, V. Reiner, Shifted set families, degree sequences, and plethysm, *Electron. J. Combin.*, **15** (2008) R14 (pp. 35).

[52] C. J. Klivans, K. L. Nyman, B. E. Tenner, Relations on generalized degree sequences, *Discrete Math.*, **309** (2009), 4377–4383.

[53] D. E. Knuth, *The Art of Computer Programming. Volume 3. Sorting and Searching* (second edition), Addison–Wesley, Reading.

[54] D. E. Knuth, *The Art of Computer Programming. Volume 4, Fascicle 0. Introduction to Combinatorial Algorithms and Boolean Functions.* Addison–Wesley, Upper Saddle River, 2008.

[55] H. G. Landau, On dominance relations and the structure of animal societies. III. The condition for a score sequence, *Bull. Math. Biophys.*, **15** (1953), 143–148.

[56] L. Lovász, *Combinatorial Problems and Exercises* (second edition). AMS Chelsea Publishing, Boston, 2007.

[57] B. D. McKay, X. Wang, Asymptotic enumeration of tournaments with a given score sequence, *J. Comb. Theory A*, **73** (1996), 77–90.

[58] E. S. Mahmoodian, A critical case method of proof in combinatorial mathematics, *Bull. Iranian Math. Soc.*, (8) (1978), 1L–26L.

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

[60] J. W. Moon, On the score sequence of an n-partite tournament, *Can. Math. Bull.*, **5** (1962), 51–58.

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

[62] J. W. Moon, *Topics on Tournaments*, Holt, Rinehart and Winston. New York, 1968.

[63] V. V. Nabiyev, H. Pehlivan, Towards reverse scheduling with final states of sports disciplines, *Appl. Comput. Math.*, **7** (2008), 89–106.

[64] T. V. Narayana, D. H. Bent, Computation of the mumber of tournament score sequences in round-robin tournaments, *Canad. Math. Bull.*, **7** (1964), 133–136.

[65] T. V. Narayana, R. M. Mathsen, J. Sarangi, An algorithm for generating partitions and its application, *J. Comb. Theory*, **11** (1971), 54–61.

[66] Ore, O. Studies on directed graphs. I. *Ann. Math.*, **63** (1956), 383–406.

[67] D. Pálvölgyi, Deciding soccer scores and partial orientations of graphs, *Acta Univ. Sapientiae, Math.*, **1** (2009), 35–42.

[68] A. N. Patrinos, S. L. Hakimi, Relations between graphs and integer-pair sequences, *Discrete Math.*, **15** (1976), 347–358.

[69] G. Pécsy, L. Szűcs, Parallel verification and enumeration of tournaments, *Stud. Univ. Babeş-Bolyai, Inform.*, **45** (2000), 11–26.

[70] S. Pirzada, M. Siddiqi, U. Samee, On mark sequences in 2-digraphs, *J. Appl. Math. Comput.*, **27** (2008), 379–391.

[71] S. Pirzada, On imbalances in digraphs, *Kragujevac J. Math.*, **31** (2008), 143–146.

[72] S. Pirzada, M. Siddiqi, U. Samee, Inequalities in oriented graph scores. II, *Bull. Allahabad Math. Soc.*, **23** (2008), 389–395.

[73] S. Pirzada, M. Siddiqi, U. Samee, On oriented graph scores, *Mat. Vesnik*, **60** (2008), 187–191.

[74] S. Pirzada, G. Zhou, Score sequences in oriented k-hypergraphs, *Eur. J. Pure Appl. Math.*, **1** (2008), 10–20.

[75] S. Pirzada, Degree sequences in multi hypertournaments, *Applied Math. – J. Chinese Univ.*, **24** (2009), 350–354.

[76] S. Pirzada, G. Zhou, On k-hypertournament losing scores, *Acta Univ. Sapientiae, Informatica*, **2** (2010), 5–9.

[77] K. B. Reid, Score sets for tournaments. *Congr. Numer.*, **21** (1978), 607–618.

[78] K. B. Reid, Tournaments: Scores, kings, generalizations and special topics, *Congr. Numer.*, **115** (1996), 171–211.

[79] K. B. Reid, C. Q. Zhang, Score sequences of semicomplete digraphs, *Bull. Inst. Combin. Appl.*, **24** (1998), 27–32.

[80] K. B. Reid, Tournaments. In *Handbook of Graph Theory* (ed. J. L. Gross, J. Yellen), CRC Press, Boca Raton, 2004.

[81] Ø. J. Rødseth, J. A. Sellers, H. Tverberg, Enumeration of the degree sequences of non-separable graphs and connected graphs, *European J. Comb.*, **30** (2009), 1309–1317.

[82] F. Ruskey, F. R. Cohen, P. Eades, A. Scott, Alley CATs in search of good homes, *Congr. Numer.*, **102** (1994), 97–110.

[83] H. J. Ryser, Matrices of zeros and ones in combinatorial mathematics, In *Recent Advances in Matrix Theory*, University of Wisconsin Press, Madison, 1964. pp. 103–124.

[84] J. K. Senior, Partitions and their representative graphs, *Amer. J. Math.*, **73** (1951), 663–689.

[85] G. Sierksma, H. Hoogeveen, Seven criteria for integer sequences being graphic, *J. Graph Theory* **15** (1991), 223–231.

[86] P. K. Stockmeyer, The falsity of the reconstruction conjecture for tournaments, *J. Graph Theory*, **1** (1977), 19–25.

[87] P. K. Stockmeyer, Erratum to: "The falsity of the reconstruction conjecture for tournaments", *J. Graph Theory*, **62** (2009), 199–200.

[88] L. A. Székely, L. H. Clark, R. C. Entringer. An inequality for degree sequences, *Discrete Math.*, **103** (1992), 293–300.

[89] C. Thomassen, Landau's characterization of tournament score sequences, In *The Theory and Applications of Graphs*, John Wiley & Sons, 1981, pp. 589–591.

[90] A. Tripathi, S. Vijay, A note on a theorem Erdős and Gallai, *Discrete Math.*, **265** (2003), 417–420.

[91] A. Tripathi, S. Vijay, On the least size of a graph with a given degree set, *Discrete Appl. Math.*, **154** (2006), 2530–2536.

[92] A. Tripathi, S. Vijay, A short proof of a theorem on degree sets of graphs, *Discrete Appl. Math.*, **155** (2007), 670–671.

[93] A. Tripathi, H. Tyagi, A simple criterion on degree sequences of graphs, *Discrete Appl. Math.*, **156** (2008), 3513–3517.

[94] R. van den Brink, R. P. Gilles, Ranking by outdegree for directed graphs. *Discrete Math.*, **271** (2003), 261–270.

[95] R. van den Brink, R. P. Gilles, The outflow ranking method for weighted directed graphs, *European J. Op. Res.*, **193** (2009), 484–491.

[96] L. Volkmann, Degree sequence conditions for super-edge-connected oriented graphs, *J. Combin. Math. Combin. Comput.*, **68** (2009), 193–204.

[97] D. L. Wang, D. J. Kleitman, On the existence of n-connected graphs with prescribed degrees ($n \geq 2$), *Networks*, **3** (1973), 225–239.

[98] C. Wang, G. Zhou, Note on the degree sequences of k-hypertournaments, *Discrete Math.* **308** (2008), 2292–2296.

[99] T. G. Will, H. Hulett, Parsimonious multigraphs, *SIAM J. Discrete Math.*, **18** (2004), 241–245.

[100] K. J. Winston, D. J. Kleitman, On the asymptotic number of tournament score sequences, *J. Combin. Theory Ser. A*, **35** (1983), 208–230.

[101] G. Zhou, T. Yao, K. Zhang, On score sequences of k-hypertournaments, *Eur. J. Comb.*, **21** (2000), 993–1000.

[102] G. Zhou, S. Pirzada, Degree sequence of oriented k-hypergraphs, *J. Appl. Math. Comput.*, **27** (2008), 149–158.

[103] T. X. Yao, On Reid conjecture of score sets for tournaments, *Chinese Sci. Bull.*, **34** (1989), 804–808.

[104] J-H. Yin, G. Chen, J. R. Schmitt, Graphic sequences with a realization containing a generalized friendship graph, *Discrete Math.*, **308** (2008), 6226–6232.

Received: March 5, 2010; Revised: April 18, 2010