A Consideration of a Score Sequence Problem of a Tournament

A Consideration of a Score Sequence Problem of a Tournament

Masaya TAKAHASHI (Department of Office Information Systems, Fukuoka Jr. College of Technology)

SUMMARY

A sequence of nonnegative integers $S=(s_1, s_2, \cdots, s_n)$ is a *score sequence of a k-tournament* if, for some positive integer k, there is a directed graph with vertices $v_1, v_2, \cdots, v_n$ such that $deg^+(v_j)=s_j$ and $deg^-(v_j)=k(n-1)-s_j$ for each $j=1, 2, \cdots, n$. The *score sequence problem of a k-tournament* is: Given some positive integer k and a sequence of nonnegative integers, determine whether it is a score sequence of a k-tournament or not. In this paper, we consider the score sequence problem of a tournament and of a k-tournament, and give efficient algorithms.

Key words : *tournament, k-tournament and score sequence*

1. Introduction

Let T be a directed graph. T is a k-tournament if and only if, for some positive integer k and some nonnegative integer $k' \leqq k$, T has k' edges (u, v) ((u, v) denotes the edge from u to v) and $k-k'$ edges (v, u) for any vertex pair of $u, v \in T$. A sequence of nonnegative integers $S=(s_1, s_2, \cdots, s_n)$ is a *score sequence of a k-tournament* if there is a k-tournament with vertices $v_1, v_2, \cdots, v_n$ such that $deg^+(v_j)=s_j$ and $deg^-(v_j)=k(n-1)-s_j$ for each $j=1, 2, \cdots, n$ ($deg^+(v_j)$ and $deg^-(v_j)$ are the outdegree and indegree of v_j respectively). The *score sequence problem of a k-tournament* is: Given some positive integer k and a sequence of nonnegative integers $S=(s_1, s_2, \cdots, s_n)$, determine whether S is a score sequence of a k-tournament or not. The k=1 case of the score sequence of a k-tournament was considered by Landau[10]. The graphical degree sequence problems and the variations of them have been considered by Havel[9], Erdös and Gallai[4], Hakimi[6], Menon[11], Takahashi, Imai and Asano[1, 12] and others[2, 3, 8].

In this paper, we consider the score sequence problem of a tournament and of a k-tournament, and give efficient algorithms.

Received Sept. 25, 1995

2. Score Sequence Problem of a Tournament

In this section, we consider the score sequence problem of a tournament and present an efficient algorithm. We first recall the previous results. Landau[10] gave Proposition 2.1 and Behzad, Chartrand and Foster[2] gave Proposition 2.2 in the following. Their proofs can be found in a standard book of graph theory[2].

Proposition 2.1. Let $S=(s_1, s_2, \cdots, s_n)$ be a sequence of nonnegative integers with $s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq n-1$. Then S is a score sequence of a tournament if and only if

$$\sum_{j=1}^{t} s_j \geqq t(t-1)/2$$

for each $j=1, 2, \cdots, n$, with equality holding for

$t=n$.

Proposition 2.2. Let $S=(s_1, s_2, \cdots, s_n)$ be a sequence of nonnegative integers with $s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq n-1$ and let $U=(u_1, u_2, \cdots, u_{n-1})$ be a sequence of nonnegative integers obtained from S by setting $u_j=s_j$ $(j=1, 2, \cdots, s_n)$ and $u_j=s_j-1(j=s_n+1, \cdots, n-1)$. Then S is a score sequence of a tournament if and only if U is a score sequence of a tournament.

Based on Proposition 2.1, we can determine whether $S=(s_1, s_2, \cdots, s_n)$ is a score sequence of a tournament or not in $O(n)$ time. Furthermore, if S is a score sequence of a tournament and $s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq n-1$, then a tournament with S as a score sequence can be obtained in $O(n^2)$ time based on Proposition 2.2.

One drawback to use Proposition 2.2 is that $u_1 \leqq u_2 \leqq \cdots \leqq u_{n-1}$ does not always hold in U. Thus we have to sort again to use Proposition 2.2 respectively. By the result of Takahashi, Imai and Asano[12], we can modify Proposition 2.2 to avoid sorting in the following.

Proposition 2.3. Let $S=(s_1, s_2, \cdots, s_n)$ be a sequence of nonnegative integers with $0 \leqq s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq n-1$ and let $U=(u_1, u_2, \cdots, u_{n-1})$ be defined by using $t=s_n+1$, $x=\min\{j \mid s_j=s_t\}$ and $y=\max\{j \mid s_j=s_t \text{ and } j \leqq n-1\}$ as follows.

$$u_j=\begin{cases} s_j-1 & \text{if } x \leqq j \leqq y-t+x \text{ or } y+1 \leqq j \leqq n-1, \\ s_j & \text{if } 1 \leqq j \leqq x-1 \text{ or } y-t+x+1 \leqq j \leqq y. \end{cases}$$

Then S is a score sequence of a tournament if and only if U is a score sequence of a tournament. Furthermore, $0 \leqq u_1 \leqq u_2 \leqq \cdots \leqq u_{n-1} \leqq n-2$.

This proposition can be proved by the similar argument as in the proof of Proposition 2.2, and we will omit a proof here. We can obtain an algorithm for constructing a tournament based on Proposition 2.3. However, a tournament has $n(n-1)/2$ edges, and thus the edge-addition takes $O(n^2)$ time. Thus the algorithm also takes $O(n^2)$ time for constructing a tournament.

In the following, we present the algorithm. In the algorithm, L is initialized $L=\{j \mid s_{j-1}<s_j, j=2, 3, \cdots, n\} \cup \{0, 1, n\}$ and represented by a doubly-linked list and $pre[j]<j<suc[j]$ for each $j \in L$, where pre[j] and suc[j] denote the previous element and the next element of $j \in L$. Note that $U=(u_1, u_2, \cdots, u_n)$ is initialized $U=S$ and then maintained to satisfy $0 \leqq u_1 \leqq u_2 \leqq \cdots \leqq u_h$, $u_{h+1}=\cdots=u_n=0$ and

$$\sum_{j=1}^{h} s_j=h(h-1)/2$$

for each $h=2, 3, \cdots, n$. L is maintained to satisfy $L=\{j \mid u_{j-1}<u_j\} \cup \{0, 1\}$. Thus $u_{pre[j]}=u_{pre[j]+1}=\cdots=u_{j-1}<u_j=u_{j+1}=\cdots=u_{suc[j]-1}$ for each $j \in L$.

Algorithm CT.
Begin
  $L:=\{0, 1\}$; **For** $j:=1$ **to** n **do** $u_j:=s_j$;
  For $j:=2$ **to** n **do**
    If $s_{j-1}<s_j$ **then**
      insert j into L as the last element of L;
  For $h:=n$ **downto** 2 **do begin**
    If h is not in L **then**
      insert h into L as the last element of L;
      Add-edge(h); delete h from L **end**
End.

Procedure Add-edge(h).
Begin
  q:=the last element of L;
  While $u_h<q-1$ **do** $q:=pre[q]$;
  For $j:=suc[q]$ **to** $h-1$ **do begin**
    add edge (v_j, v_h); $u_j:=u_j-1$ **end**;
  $q_{new}:=q-u_h+suc[q]-2$;
  For $j:=q_{new}+1$ **to** $suc[q]-1$ **do** add edge (v_h, v_j);
  For $j:=q$ **to** q_{new} **do begin**
    add edge (v_j, v_h); $u_j:=u_j-1$ **end**;
  For $j:=1$ **to** $q-1$ **do** add edge (v_h, v_j);
  If $u_h>q-1$ **then begin**
    insert $q_{new}+1$ into L between q and suc[q];
    If $u_{suc[q]-1}=u_{suc[q]}$ **then**
      delete suc[q] from L **end**;
  If $u_{q-1}=u_q$ **then** delete q from L
End.

It is easy to see that Algorithm CT correctly constructs a tournament T with S as a score sequence and that it takes $O(n^2)$ time. Thus we have the following theorem.

Theorem 1. For a sequence of nonnegative integers $S=(s_1, s_2, \cdots, s_n)$, we can determine in

$O(n)$ time whether S is a score sequence of a tournament or not[2]. Furthermore, if $S=(s_1, s_2, \cdots, s_n)$ is a score sequence of a tournament and $0 \leqq s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq n-1$ then a tournament T with S as a score sequence can be constructed in $O(n^2)$ time.

3. Score Sequence Problem of a k-Tournament

In this section, we consider the score sequence problem of a k-tournament. First we can obtain the following proposition.

Proposition 3.1. Let $S=(s_1, s_2, \cdots, s_n)$ be a sequence of nonnegative integers with $s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq k(n-1)$. Then S is a score sequence of a k-tournament if and only if

$$\sum_{j=1}^{t} s_j \geqq kt(t-1)/2$$

for each $j=1, 2, \cdots, n$, with equality holding for $t=n$.

This proposition can be proved by the similar argument as in the proof of Proposition 2.1[2, 9], and we will omit a proof here. Based on Proposition 3.1, we can determine whether S is a score sequence of a k-tournament or not in $O(n)$ time.

Next we present an algorithm for actually constructing a k-tournament for a given score sequence. Then the algorithm takes at least $O(n^2)$ time, since a k-tournament has at least $n(n-1)/2$ kinds of edges (i.e., $|(v_j, v_q)|+|(v_q, v_j)|=k$ for each $1 \leqq j < q \leqq n$). Thus we present the algorithm takes $O(n^2)$ time in the following. We can obtain the following proposition.

Proposition 3.2. Let $S=(s_1, s_2, \cdots, s_n)$ be a sequence of nonnegative integers with $0 \leqq s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq k(n-1)$ and let $U=(u_1, u_2, \cdots, u_{n-1})$ be defined by the following algorithm. Then S is a score sequence of a k-tournament if and only if U is a score sequence of a k-tournament.

Algorithm k T.

Step1: $R_n := k(n-1)-s_n$.

Step2: **If** $n \neq 3$ **then do** the following.

For $j := n-1$ **downto** 1 **do**

$h := \min\{s_j, k, R_{j+1}\}$, $u_j := s_j - h$ and $R_j := R_{j+1} - h$.

Step3: **If** $n=3$ **then do** the following.

For $j := 1$ **to** $n-1$ **do**

$h := \min\{s_j, k, R_{j+1}\}$, $u_j := s_j - h$ and $R_j := R_{j+1} - h$.

Proof. The sufficiency is almost trivial. Let $f_{jn} = s_j - u_j$ for each $j=1, 2, \cdots, n-1$. If U is a score sequence of a k-tournament and H is a k-tournament with U as the score sequence, then the k-tournament T obtained from H by adding f_{jn} edges (v_j, v_n) and $k-f_{jn}$ edges (v_n, v_j) for each $j=1, 2, \cdots, n-1$, has S as a score sequence.

The necessity can be obtairied as follows. Let T be a k-tournament with S as a score sequence. If T contains f_{jn} edges (v_j, v_n) and $k-f_{jn}$ edges (v_n, v_j) for each $j=1, 2, \cdots, n-1$, then the k-tournament H obtained from T by deleting v_n has U as a score sequence. If $n=2$ then this argument always holds. Hence we assume that $n \geqq 3$ and that T contains g_{tn} edges (v_t, v_n) with $g_{tn} < f_{tn}$ for some t with $1 \leqq t \leqq n-1$. Then we have $R_n > 0$ and T contains g_{hn} edges (v_h, v_n) with $g_{hn} > f_{hn}$ for some h with $1 \leqq h \leqq n-1$.

Let $g_{jn} = |\{(v_j, v_n)\} \in T|$ for each $j=1, 2, \cdots, n-1$. Let $X=\{v_j \mid g_{jn} < f_{jn}, 1 \leqq j \leqq n-1\}$, $W=\{v_j \mid g_{jn} = f_{jn}, 1 \leqq j \leqq n-1\}$ and $Y=\{v_j \mid g_{jn} > f_{jn}, 1 \leqq j \leqq n-1\}$. Then T has at least one edge (v_h, v_n) for a vertex $v_h \in Y$ since $g_{hn} > f_{hn} \geqq 0$, and has at least one edge (v_n, v_t) for a vertex $v_t \in X$ since $k - g_{tn} \geqq f_{tn} - g_{tn} > 0$. We consider two cases.

Case 1: Suppose $n \geqq 4$. Let $z=\{j \mid R_j = 0$ and $R_{j+1} > 0 (1 \leqq j \leqq n-1)\}$. Then we have $f_{jn} = \min\{s_j, k\}$ for each $j=z+1, \cdots, n-1$, $0 \leqq f_{zn} \leqq \min\{s_z, k\}$ and $f_{jn}=0$ for each $j=1, 2, \cdots, z-1$, and thus we have $X \subseteq \{v_j \mid j=z, z+1, \cdots, n-1\}$ and $Y \subseteq \{v_j \mid j=1, 2, \cdots, z\}$, where $X \cup Y = \Phi$. Then we can prove that T has a directed path from v_t to v_h without v_n for some two vertices $v_t \in X$ and $v_h \in Y$ as follows. We can assume $0 \leqq s_{n-1} - s_1 \leqq 1$ because

$$\sum_{v_t \in X} \deg^+(v_t) = \sum_{v_t \in X} s_t$$

is minimum. Thus we have

$$\left(\sum_{j=1}^{n-1} s_j\right)/(n-1) = kn/2 - s_n/(n-1)$$

and

$\sum_{v_t \in X} \deg^+(v_t) = \sum_{v_t \in X} s_t \geqq (kn/2 - s_n/(n-1))|X|$.

Furthermore, we assume that $T - v_n$ does not contain an edge (v_t, v') for any two vertices $v_t \in X$ and $v' \notin X$. Then we have

$\sum_{v_t \in X} \deg^+(v_t) = k|X|(|X|-1)/2 + \sum_{v_t \in X} g_{tn}$

and

$\sum_{v_t \in X} g_{tn} \geqq (kn/2 - s_n/(n-1))|X| - k|X|(|X|-1)/2$

$= k|X|(n-|X|+1)/2 - (|X|/(n-1))\, s_n$.

Hence we obtain $g' = k(n-|X|+1)/2 - s_n/(n-1) \geqq k(n-|X|-1)/2$, where g' is the average of g_{tn}. We consider two cases.

Case 1-A: If $n-|X|-1 \geqq 2$ then $g' \geqq k$ holds and contradicts $g_{tn} < f_{tn} \leqq k$.

Case 1-B: Suppose $n-|X|-1=1$. Then we have $|W|=0$, $|Y|=1$ and $g' \geqq k/2$. Let $Y=\{v_h\}$. Since $s_n \geqq s_h$ and $T-v_n$ does not contain an edge (v_t, v') for any two vertices $v_t \in X$ and $v' \notin X$, we have $\deg^+(v_h) = s_h = g_{hn} + k(n-2)$ and $s_n \geqq g_{hn} + k(n-2)$. Thus we obtain

$\sum_{v_t \in X} g_{tn} + g_{hn} = k(n-1) - s_n \leqq k(n-1) - g_{hn} - k(n-2)$

$= k - g_{hn}$, and

$\sum_{v_t \in X} g_{tn} \leqq k - 2g_{hn}$.

Since $|X| = n-2$ and $k/2 \leqq g'$, we obtain

$k(n-2)/2 \leqq \sum_{v_t \in X} g_{tn} \leqq k - 2g_{hn}$

and $g_{hn} \leqq 2k - kn/2$. Thus $g_{hn} \leqq 0$ holds and contradicts $0 \leqq f_{hn} < g_{hn}$, since $n \geqq 4$.

Hence, by Case 1-A and 1-B, T-v_n contains an edge (v_t, v') for some two vertices $v_t \in X$ and $v' \notin X$. We consider two cases.

Case 1-C: If $v' \in Y$ then T has a directed path from v_t to v_h without v_n for some two vertices $v_t \in X$ and $v_h \in Y$, and thus, has a directed cycle containing (v_h, v_n) and (v_n, v_t) for some two vertices $v_t \in X$ and $v_h \in Y$.

Case 1-D: Suppose $v' \in W$. Let $W' = \{v' | (v_t, v'), v_t \in X \text{ and } v' \in W\}$. By setting $X := X \cup W'$ and $W := W - W'$ and repeating the argument above, we can finally obtain that T has a directed path from v_t to v_h without v_n for some two vertices $v_t \in X$ and $v_h \in Y$, and thus, has a directed cycle containing (v_h, v_n) and (v_n, v_t) for some two vertices $v_t \in X$ and $v_h \in Y$.

Let E_1 be an edge set of the directed cycle. Let $E_2 = \{(u, v) | (v, u) \in E_1\}$. Then $T' = (T \cup E_2) - E_1$ has also $S = (s_1, s_2, \cdots, s_n)$ as a score sequence. By setting $T := T'$ and repeating the argument above, we can finally obtain a k-tournament T which contains f_{jn} edges (v_j, v_n) for each $j = 1, 2, \cdots, n-1$.

Case 2: Suppose $n=3$. Then we have $|f_{12} - g_{12}| = |f_{13} - g_{13}|$ and $s_1 \leqq k$. We consider two cases.

Case 2-A: Suppose $R_3 \leqq s_1$. Since $f_{13} = R_3$ and $f_{12} = 0$, we have $f_{13} > g_{13}$, $f_{12} < g_{12}$ and $|(v_1, v_2)| = s_1 - g_{13} = s_1 - f_{13} + (f_{13} - g_{13})$. Furthermore, we have $|(v_1, v_2)| \geqq f_{13} - g_{13}$ since $s_1 - f_{13} \geqq 0$. Let T' be a k-tournament obtained as follows:

(1) Delete $f_{13} - g_{13}$ edges (v_3, v_1), $g_{12} - f_{12}$ edges (v_2, v_3) and $f_{13} - g_{13}$ edges (v_1, v_2) from T, and

(2) Add $f_{13} - g_{13}$ edges (v_1, v_3), $g_{12} - f_{12}$ edges (v_3, v_2) and $f_{13} - g_{13}$ edges (v_2, v_1) to the directed graph obtained by (1).

Case 2-B: Suppose $R_3 > s_1$. Since $f_{13} = s_1$ and $f_{12} = \min\{k, R_3 - s_1\}$, we have $f_{13} > g_{13}$, $f_{12} < g_{12}$ and $|(v_1, v_2)| = f_{13} - g_{13}$. Furthermore, we have $g_{12} > f_{13} - g_{13}$, since $g_{12} = \min\{k, R_3 - s_1\} + (f_{13} - g_{13})$, $R_3 - s_1 > 0$ and $\min\{k, R_3 - s_1\} > 0$. Let T' be a k-tournament obtained by (1) and (2) above.

Then, by Case 2-A and 2-B, T' also has $S = (s_1, s_2, \cdots, s_n)$ as a score sequence of a k-tournament. Set $T := T'$.

Hence, by the argument above, the k-tournament H obtained from T by deleting v_n has U as a score sequence. ■

Algorithm kT takes $O(n)$ time. However $u_1 \leqq u_2 \cdots \leqq u_{n-1}$ does not always hold in U. (If $n=3$ then $u_1 \leqq u_2$ always holds in U.) Thus we can obtain a k-tournament in $O(n^2)$ time for a given score sequence if we can sort in $O(n)$ time to use Algorithm kT recursively.

We first modify Algorithm kT to sort in $O(n)$ time.

Algorithm kT-1.

Step1: $R := k(n-1) - s_n$ and,
 If $R = 0$ **then** $z_1 := n$ and $z_2 := n-1$.

Step2: If $n \geqq 4$ **then do** the following.

For $j:=n-1$ **downto** 1 **do**
the following (a) through (c).
(a) $h:=\min\{s_j, k, R\}$, $u_j:=s_j-h$ and $R':=R-h$.
(b) **If** $R>0$ **and** $R'=0$ **then** $z_1:=j$ and $z_2:=j-1$.
(c) $R:=R'$.

Step3: **If** $n=3$ **then do** the following.
For $j:=1$ **to** $n-1$ **do**
$h:=\min\{s_j, k, R\}$, $u_j:=s_j-h$ and $R:=R-h$.

Step4: **If** $n=2$ **then** $u_1:=R-s_1(=0)$.

Then we can obtain the following proposition.

Proposition 2.5. Suppose that $n\geqq 4$. Let p be any permutation of $\{1, 2, \cdots, n-1\}$ and p' be any permutation of $\{z_1, z_1+1, \cdots, n-1\}$. Let $U'=(u_{p(1)}, u_{p(2)}, \cdots, u_{p(n-1)})$ be defined by the following algorithm. Then $u_{p(1)}\leqq u_{p(2)}\leqq\cdots\leqq u_{p(n-1)}$, where z_1 and z_2 are integers defined by Step1 or Step2-(b) of Algorithm kT-1.

Algorithm S.

Step1: $p'(j):=j$ for each $j=z_1, z_1+1, \cdots, n-1$ and $h:=z_1$.

Step2: **While** $u_{p'(h)}>u_{p'(h+1)}$ **do**
swap $\{p'(h), p'(h+1)\}$ and $h:=h+1$.

Step3: $h_1:=n-1$, $h_2:=z_2$ and $z:=n-1$.

Step4: **While** $h_1\geqq z_1$ **and** $h_2\geqq 1$ **do**
the following (a) and (b).
(a) **If** $u_{p'(h_1)}\geqq u_{h_2}$ **then**
$p(z):=p'(h_1)$ and $h_1:=h_1-1$
else $p(z):=h_2$ and $h_2:=h_2-1$.
(b) $z:=z-1$.

Step5: **If** $h_1\geqq z_1$ **then**
For $j:=h_1-z_2$ **downto** 1 **do** $p(j):=p'(z_2+j)$
else if $h_2\geqq 1$ **then**
For $j:=h_2$ **downto** 1 **do** $p(j):=j$.

Proof. By Algorithm kT-1, we have $u_j=s_j-k$ or $u_j=0$ for each $j=z_1+1, \cdots, n-1$, since $R>h=\min\{s_j, k\}$. Thus $u_{z_1+1}\leqq u_{z_1+2}\leqq\cdots\leqq u_{n-1}$. Furthermore, we have $u_j=s_j$ for each $j=1, 2, \cdots, z_2$, since $R=0$. Thus $u_1\leqq u_2\leqq\cdots\leqq u_{z_2}$. Hence we obtain $u_{p(1)}\leqq u_{p(2)}\leqq\cdots\leqq u_{p(n-1)}$. ■

Based on Proposition 2.4 and 2.5, we can obtain the following iterative algorithm CkT for constructing a k-tournament T with S as a score sequence of a k-tournament.

Let K_n be the complete directed graph with vertex set $V=\{v_1, v_2, \cdots, v_n\}$ and let $N=(K_n, cap)$ be the weighted graph defined by the capacity function $cap(e)=cap(e')=k$ for $e=(v_j, v_q)$, $e'=(v_q, v_j)$, $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$. Since S is a score sequence of a k-tournament, N has a weight of value

$$m=\sum_{j=1}^{n} s_j.$$

For a weight w of N of value m, we create w(e) copies of an edge $e=(v_j, v_q)$ and $w(e')$ copies of an edge $e'=(v_q, v_j)$, $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$. Then the complete directed graph H_w obtained in this way has S as a score sequence of a k-tournament. To obtain a k-tournament, we obtain a maximum weight w of N.

Let h be any integer in $\{1, 2, \cdots, n\}$. Let p_1, p'_1 and p_2 be any permutations of $\{1, 2, \cdots, h\}$ and $U(h)=(u_{p_1(1)}, u_{p_1(2)}, \cdots, u_{p_1(n)})$ be a sequence of nonnegative integers with

$$\sum_{j=1}^{h} u_{p_1(j)}=kh(h-1)/2,$$

$0\leqq u_{p_1(1)}\leqq u_{p_1(2)}\leqq\cdots\leqq u_{p_1(h)}\leqq k(h-1)$ and $u_{p_1(h+1)}=\cdots=u_{p_1(n)}=0$. In the algorithm, h is initialized $h=n$ and U(h) is initialized $U(h)=S$. We assume $w(e)=0$ for $e\in E(K_n)$ at the beginning, where $E(K_n)$ is an edge set of K_n.

Algorithm CkT.

Begin
$U(n):=S$; **For** $j:=1$ **to** n **do** $p_1(j):=j$;
For $h:=n$ **downto** 4 **do begin**
$R:=k(h-1)-u_{p_1(h)}$;
If $R=0$ **then begin** $z_1:=h$; $z_2:=h-1$ **end**;
For $j:=h-1$ **downto** 1 **do begin**
$t:=\min\{u_{p_1(j)}, k, R\}$;$u_{p_1(j)}:=u_{p_1(j)}-t$;
$u_{p_1(h)}:=u_{p_1(h)}-(k-t)$;$R':=R-t$;
$w((v_{p_1(j)}, v_{p_1(h)})):=t$;
$w((v_{p_1(h)}, v_{p_1(j)})):=k-t$;
If $R>0$ **and** $R'=0$ **then begin**
$z_1:=j$; $z_2:=j-1$ **end**;
$R:=R'$ **end**;
For $j:=z_1$ **to** $n-1$ **do** $p'_1(j):=p_1(j)$; $t:=z_1$;
While $u_{p'_1(t)}>u_{p'_1(t+1)}$ **do begin**
swap $\{p'_1(t), p'_1(t+1)\}$; $t:=t+1$ **end**;

$t_1 = h-1$; $t_2 := z_2$; $z := h-1$;
While $t_1 \geqq z_1$ and $t_2 \geqq 1$ do begin
If $u_{p'_1(t_1)} \geqq u_{p_1(t_2)}$ then begin
$p_2(z) := p'_1(t_1)$; $t_1 := t_1 - 1$ end
else begin $p_2(z) := p_1(t_2)$; $t_2 := t_2 - 1$ end;
$z := z-1$ end;
If $t_1 \geqq z_1$ then
For $j := t_1 - z_2$ downto 1 do $p_2(j) := p'_1(z_2 + j)$
else if $t_2 \geqq 1$ then
For $j := t_2$ downto 1 do $p_2(j) := p_1(j)$;
$p_1 := p_2$ end;
$R := 2k - u_{p_1(3)}$;
For $j := 1$ to 2 do begin
$t := \min\{u_{p_1(j)}, k, R\}$; $u_{p_1(j)} := u_{p_1(j)} - t$;
$u_{p_1(3)} := u_{p_1(3)} - (k-t)$; $w((v_{p_1(j)}, v_{p_1(3)})) := t$;
$w((v_{p_1(3)}, v_{p_1(j)})) := k - t$; $R := R - t$ end;
$w((v_{p_1(1)}, v_{p_1(2)})) := u_{p_1(1)}$;
$w((v_{p_1(2)}, v_{p_1(1)})) := u_{p_1(2)}$; $u_{p_1(1)} := 0$;
$u_{p_1(2)} := 0$
End.

It is easy to see that Algorithm CkT correctly constructs a k-tournament T with S as a score sequence and that it takes $O(n^2)$ time. Thus we have the following theorem.

Theorem 2. For a sequence of nonnegative integers $S = (s_1, s_2, \cdots, s_n)$, we can determine in $O(n)$ time whether S is a score sequence of a k-tournament or not. Furthermore, if $S = (s_1, s_2, \cdots, s_n)$ is a score sequence of a k-tournament and $0 \leqq s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq k(n-1)$ then a k-tournament T with S as a score sequence can be constructed in $O(n^2)$ time.

4. Concluding Remarks

We have considered the score sequence problem of a tournament and of a k-tournament, and given optimal algorithms. Especially, we have found that there is an intimate relation between the graphical degree sequence problem[12] and the score sequence problem of a tournament. Furthermore, we have presented an algorithm for constructing a tournament having a given score sequence without sorting. However, the algorithm, which is presented in this paper, for constructing a k-tournament having a given score sequence contains sorting.

For our further investigations, we will consider an algorithm for constructing a k-tournament having a given score sequence without sorting.

References

[1] Asano, T., *An O(nloglogn) Time Algorithm for Constructing a Graph of Maximum Connectivity with Prescribed Degrees*, Technical Report, ISETR93-02, Department of Information and System Engineering, Chuo University, 1993.

[2] Behzard, M., Chartrand, G. and Foster, L.L. -, "*Graphs and Digraphs*," Prindle, Weber and Schmidt, 1979.

[3] Chen, Y.K., "*Applied Graph Theory* (Second Revised Edition)," North-Holland, 1976.

[4] Erdös, P. and Gallai, T., *Graphs with Prescribed Degrees of Vertices* (Hungarian), Mat. Lapok, Vol. 11, pp. 264-274, 1960.

[5] Ford, Jr, L.R. and Fulkerson, D.R., "*Flows in Networks*," Princeton University Press, 1962.

[6] Hakimi, S.L., *On the Realizability of a Set of Integers as Degrees of the Vertices of a Graph*, J. SIAM Appl. Math., Vol. 10, pp. 496-506, 1962.

[7] Hakimi, S.L. and Moser, L., *The Theory of Round Robin Tournaments*, Amer. Math. Monthly 73, pp. 231-246, 1966.

[8] Harary, F., "*Graph Theory*," Addison-Wesley, 1969.

[9] Havel, V., *A Remark on the Existence of Finite Graph* (Hungarian), Časopis Pěst., Mat., Vol. 80, pp. 477-480, 1955.

[10] Landau, H.G., *On Dominance Relations and the Structure of Animal Societies, III., The Condition for a Score Structure*, Bulletin for Mathematical Biophysics, Vol. 15, pp. 143-148, 1953.

[11] Menon, V.V., *On the Existence of Trees with*

Given Degrees, Sankhyã, Series A, 26 (1964), pp. 63-68.

[12] Takahashi, M., Imai, K. and Asano, T., *Graphical Degree Sequence Problems*, IEICE Trans. Fundamentals., Vol. E77-A. No. 3 (March), pp. 546-552, 1994.