Several Approaches of Score Sequence Problems of k-Tournaments

福岡工業大学研究論集 Res. Bull. Fukuoka Inst. Tech., Vol. 29 No. 1 (1996) 111—119

Several Approaches of Score Sequence Problems of k-Tournaments

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 some approaches of the score sequence problem of a k-tournament, and give efficient algorithms.

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

1. Introduction

Let T be a directed graph and k be a positive integer. T is a *k-tournament* if and only if T satisfies $|(u, v)| \geqq 0, |(v, u)| \geqq 0$ and $|(u, v)|+|(v, u)|=k$ for any vertex pair u, $v \in T$, where (u, v) and (v, u) denote the edge from u to v and the edge from v to u. 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 score sequence problem of an 1-tournament was considered by Landau[10], and the score sequence problem of a k-tournament was first considered by Takahashi[12]. 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, 13] and others[2, 3, 8].

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

Received May 29, 1996

2. Score Sequence Problem of a k-Tournament

In this section, we recall the previous results of the score sequence problem of a k-tournament. 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 an

1-tournament if and only if

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

for each $t=1, 2, \cdots, n$, 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 an 1-tournament if and only if U is a score sequence of an 1-tournament.

Furthermore, Takahashi[12] modified Proposition 2.2 to avoid sorting as follows.

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 as follows.

(1) If $s_n=n-1$ then $u_j=s_j$ for each $j=1, 2, \cdots, n-1$,

(2) If $s_n<n-1$ then

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

where $t=s_n+1$, $x=min\{j \mid s_j=s_t\}$ and $y=max\{j \mid s_j=s_t \text{ and } j \leqq n-1\}$. Then S is a score sequence of an 1-tournament if and only if U is a score sequence of an 1-tournament. Furthermore, $0 \leqq u_1 \leqq u_2 \leqq \cdots \leqq u_{n-1} \leqq n-2$.

Based on Proposition 2.1, we can determine whether $S=(s_1, s_2, \cdots, s_n)$ is a score sequence of an 1-tournament or not in $O(n)$ time. Furthermore, if S is a score sequence of an 1-tournament and $s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq n-1$, then an 1-tournament with S as a score sequence can be obtained in $O(n^2)$ time based on Proposition 2.2 and 2.3.

Next Takahashi[12] also gave Proposition 2.4 and 2.5 in the following.

Proposition 2.4. 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 $t=1, 2, \cdots, n$, with equality holding for $t=n$.

Proposition 2.5. 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 *kT*-1.

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 $R_j:=R_{j+1}-h$.

Based on Proposition 2.4, we can determine whether $S=(s_1, s_2, \cdots, s_n)$ is a score sequence of a k-tournament or not in $O(n)$ time. Furthermore, if S is a score sequence of a k-tournament and $s_1 \leqq s_2 \leqq \cdots \leqq s_n \leqq k(n-1)$, then a k-tournament with S as a score sequence can be obtained in $O(n^2)$ time based on Proposition 2.5[12].

In the following, we consider other algorithms for constructing a k-tournament with S as a score sequence.

3. First Approach

Any 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 $1 \leqq j < q \leqq n$). Our algorithm takes $O(n^2)$ time and is optimal. We can assume without loss of generality that $n \geqq 2$ and obtain the following proposition.

Proposition 3.1. 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 $U_1=(u_1, u_2, \cdots, u_q)$ and $U_2=(u_{q+1}, u_{q+2}, \cdots, u_n)$ be defined from S by the following algorithm, where $1 \leqq q \leqq n-1$. Then S is a score sequence of a k-tournament if and only if U_1 and U_2 are score sequences of k-tournaments. Furthermore, $0 \leqq u_1 \leqq u_2 \leqq \cdots < u_q \leqq k(q$

$-1)$ and $0 \leqq u_{q+1} \leqq u_{q+2} \leqq \cdots \leqq u_n \leqq k(n-q-1)$.

Algorithm kT-2.

Comment: Variables x, $f_{jh}(j=1, 2, \cdots, q, h=q+1, q+2, \cdots, n)$ and $R_j(j=q+1, q+2, \cdots, n)$ described below are merely tools to be used in the following proof.

Step1: $q := max\{t \mid min\{\sum_{j=1}^{t} s_j - k \cdot t(t-1)/2\}, 1 \leqq t \leqq n-1\}$ and
$b := \sum_{j=1}^{q} s_j - k \cdot q(q-1)/2$.
$\{$(i.e., $b < \sum_{j=1}^{t} s_j - k \cdot t(t-1)/2$ for each $t = q+1, q+2, \cdots, n-1.)\}$

Step2: **If** $b<0$ **then** $b_0 := 0$ **else** $b_0 := b$.

Step3: $f_{jh} := 0$ for each $j=1, 2, \cdots, q$, $h=q+1, q+2, \cdots, n$.

Step4: $R_j := 0$ for each $j=q+1, q+2, \cdots, n$.

Step5: **If** $b_0=0$ **then** $x:=0$.

Step6: **For** $j:=1$ **to** q **do** the following (a) through (c).

(a) $z_j := min\{k, b_{j-1}\}$.

(b) $u_j := s_j - z_j$, $b_j := b_{j-1} - z_j$, $f_{jn} := f_{jn} + z_j$ and $R_n := R_n + z_j$.

(c) **If** $b_{j-1}>0$ **and** $b_j=0$ **then** $x:=j$.

Step7: $b_n := b_q$.

Step8: **For** $j:=n-1$ **downto** $q+1$ **do** the following (a) through (c).

(a) $z_j := min\{k, b_{j+1}\}$.

(b) $u_1 := u_1 - z_j$, $b_j := b_{j+1} - z_j$, $f_{1j} := f_{1j} + z_j$ and $R_j := R_j + z_j$.

(c) **If** $b_{j+1}>0$ **and** $b_j=0$ **then** $x:=j$.

Step9: **For** $j:=q+1$ **to** n **do** $u_j := s_j - k \cdot q + R_j$.

Note. Our object is to separates the sequence S to two sequences U_1 containing u_1 and U_2 containing u_n, by adding edges in $\{(v_j, v_h)$ and $(v_h, v_j) \mid j=1, 2, \cdots, q,\ h=q+1, q+2, \cdots, n\}$ such that $|(v_j, v_h)| \geqq 0$, $|(v_h, v_j)| \geqq 0$ and $|(v_j, v_h)| + |(v_h, v_j)| = k$ hold and

$$\sum_{j=1}^{q} \sum_{h=q+1}^{n} |(v_j, v_h)|\ (\geqq 0)$$

is minimum. Furthermore, if S is a score sequence of a k-tournament then $b \geqq 0$ always holds in Step2 of Algorithm kT-2 by Proposition 2.3.

Proof. We first consider the correctness of Algorithm kT-2. Suppose $b<0$. Then $b_0=0$ and $x=0$. Thus Step5, 6 and 8 are well-defined respectively, $u_j=s_j$ for each $j=1, 2, \cdots, q$, $u_j=s_j-k \cdot q$ for each $j=q+1, q+2, \cdots, n$, and $0 \leqq f_{jh} \leqq k$ for each $j=1, 2, \cdots, q$, $h=q+1, q+2, \cdots, n$. Suppose $b \geqq 0$. Then $b_0=b$. Since

$$\sum_{j=1}^{1} s_j - k \cdot 1(1-1)/2 = s_1 \geqq b_0$$

and

$$\sum_{j=1}^{n-1} s_j - k(n-1)(n-2)/2 = k(n-1) - s_n \geqq b_0,$$

we have $b_0 \leqq s_1 \leqq s_2 \leqq \cdots \leqq s_q$ and $k(n-1)-s_{q+1} \geqq k(n-1)-s_{q+2} \geqq \cdots \geqq k(n-1)-s_{n-1} \geqq k(n-1)-s_n \geqq b_0$. Hence Step 6-$(a)$ and (b) and Step 8-(a) and (b) are well-defined respectively, and $0 \leqq f_{jh} \leqq k$ for each $j=1, 2, \cdots, q$, $h=q+1, q+2, \cdots, n$. If $b_{q+1}>0$ in Step8-(c) then $b_0>k(n-1)$, and thus $s_1>k(n-1)$, $s_n \leqq k(n-1)-b_0<0$ and a contradiction. Hence we have $b_0 \leqq k(n-1)$ and $0 \leqq x \leqq n-1$, and thus Step6-(c) and Step8-(c) are well-defined. Furthermore, Step9 corresponds to that f_{jh} edges (v_j, v_h) and $k-f_{jh}$ edges (v_h, v_j) are added for each $j=1, 2, \cdots, q$, $h=q+1, q+2, \cdots, n$. Hence Algorithm kT-2 attains our object.

Next we prove Proposition 2.4. The sufficiency is almost trivial. If U_1 and U_2 are score sequences of k-tournaments and H_1 and H_2 are k-tournaments with U_1 and U_2 as the score sequences respectively, then a k-tournament T obtained from H_1 and H_2 by adding f_{jh} edges (v_j, v_h) and $k-f_{jh}$ edges (v_h, v_j) for each $j=1, 2, \cdots, q$, $h=q+1, q+2, \cdots, n$, has S as a score sequence.

The necessity can be obtained as follows. Since

$$b_0 \leqq \sum_{j=1}^{t} s_j - kt(t-1)/2$$

for each $t=1, 2, \cdots, n-1$,

$$\sum_{j=1}^{t} (s_j - u_j) \leqq b_0$$

for each $t=1, 2, \cdots, q$, with equality holding for $t=q$, and

$$\sum_{j=1}^{t} (s_j - u_j + k \cdot q - R_j) \leqq b_0$$

for each $t=q+1, q+2, \cdots, n-1$, we have

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

for each $t=1, 2, \cdots, q$, with equality holding for $t=q$, and

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

for each $t=q+1, q+2, \cdots, n$, with equality holding for $t=n$. Furthermore, it is easy to prove that $u_q \leqq k(q-1)$ and $u_n \leqq k(n-q-1)$. We can prove $0 \leqq u_1 \leqq u_2 \leqq \cdots \leqq u_q$ and $0 \leqq u_{q+1} \leqq u_{q+2} \leqq \cdots \leqq u_n$ as follows. We consider two cases.

Case 1: Suppose $0 \leqq x \leqq q$. If $x=0$ then $u_j = s_j$ for each $j=1, 2, \cdots, q$, and $u_j = s_j - k \cdot q$ for each $j=q+1, q+2, \cdots, n$, and thus $0 \leqq u_1 \leqq u_2 \leqq \cdots \leqq u_q$ and $0 \leqq u_{q+1} \leqq u_{q+2} \leqq \cdots \leqq u_n$. Assume $x \geqq 1$. Then $f_{1n} = \cdots = f_{x-1,n} = k \geqq f_{xn} \geqq 0 = f_{x+1,n} = \cdots = f_{qn}$ and $f_{1,q+1} = \cdots = f_{1,n-1} = 0$, and thus $R_{q+1} = \cdots R_{n-1} = 0$ and $R_n = f_{1n} + \cdots + f_{xn} = b_0$. Hence we have $u_1 = s_1 - f_{1n} \leqq u_2 = s_2 - f_{2n} \leqq \cdots \leqq u_q = s_q - f_{qn}$ and $u_{q+1} = s_{q+1} - k \cdot q + R_{q+1} \leqq u_{q+2} = s_{q+2} - k \cdot q + R_{q+2} \leqq \cdots \leqq u_n = s_n - k \cdot q + R_n$. Furthermore, $u_1 = s_1 - f_{1n} \geqq s_1 - b_0 \geqq 0$ and $u_{q+1} = s_{q+1} - k \cdot q + R_{q+1} \geqq s_{q+1} - k \cdot q \geqq 0$ since

$$\sum_{j=1}^{q} u_j = kq(q-1)/2 \text{ and } \sum_{j=1}^{q+1} u_j \geqq kq(q+1)/2.$$

Case 2: Suppose $q+1 \leqq x \leqq n-1$. Then $f_{1n} = \cdots = f_{qn} = k$, $f_{1,x+1} = \cdots = f_{1,n-1} = k \geqq f_{1x} \geqq 0 = f_{1,q+1} = \cdots = f_{1,x-1}$, and thus $R_{q+1} = \cdots = R_{x-1} = 0$, $R_x = f_{1x}$, $R_{x+1} = \cdots = R_{n-1} = k$ and $R_n = k \cdot q$. Hence we have $u_1 = s_1 - (f_{1n} + R_x + \cdots + R_{n-1}) \leqq u_2 = s_2 - f_{2n} \leqq \cdots \leqq u_q = s_q - f_{qh}$ and $u_{q+1} = s_{q+1} - k \cdot q + R_{q+1} \leqq u_{q+2} = s_{q+2} - k \cdot q + R_{q+2} \leqq \cdots \leqq u_n = s_n - k \cdot q + R_n$. Furthermore, $u_1 = s_1 - (f_{1n} + R_x + \cdots + R_{n-1}) \geqq s_1 - b_0 \geqq 0$ and $u_{q+1} = s_{q+1} - k \cdot q + R_{q+1} \geqq s_{q+1} - k \cdot q \geqq 0$.

By the argument above, $0 \leqq u_1 \leqq u_2 \leqq \cdots \leqq u_q \leqq k(q-1)$ and $0 \leqq u_{q+1} \leqq u_{q+2} \leqq \cdots \leqq u_n \leqq k(n-q-1)$. Hence $U_1 = (u_1, u_2, \cdots, u_q)$ and $U_2 = (u_{q+1}, u_{q+2}, \cdots, u_n)$ are score sequences of k-tournaments by Proposition 2.3. ■

Algorithm kT-2 separates a sequence S to two sequences U_1 and U_2. Thus we can obtain $u_1 = u_2 = \cdots = u_n = 0$ by using Algorithm kT-2 $n-1$ times, and can obtain the following iterative algorithm CkT-1 for constructing a k-tournament T with S as a score sequence, based on Proposition 2.4. In the following, we can assume without loss of generality that S is 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 $N=(K_n, \text{cap})$ be the complete weighted directed graph defined by the capacity function cap(e)=cap(e′)=k for each edge $e=(v_j, v_q) \in K_n$, $e'=(v_q, v_j) \in K_n$, $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$. Then we define a weight w as follows:

(1) $w(e)+w(e')=k$,

(2) $w(e)$ and $w(e')$ are nonnegative integers,

for each edge $e=(v_j, v_q) \in K_n$, $e'=(v_q, v_j) \in K_n$, $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$. Then S is a score sequence of a k-tournament if and only if

$$\sum_{q=1}^{j-1} w((v_j, v_q)) + \sum_{q=j+1}^{n} w((v_j, v_q)) = s_j$$

for each $j=1, 2, \cdots, n$, and N has a total weight

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

In fact, if H is a k-tournament with S as a score sequence, then a weight w satisfies $w(e)=|e_1|$, $w(e')=k-|e_1|=|e_1'|$ ($e_1=(v_j, v_q) \in$ H, $e=(v_j, v_q) \in K_n$, $e_1'=(v_q, v_j) \in H$, $e'=(v_q, v_j) \in K_n$, $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$) and N has a total weight m. On the other hand, if

$$\sum_{q=1}^{j-1} w((v_j, v_q)) + \sum_{q=j+1}^{n} w((v_j, v_q)) = s_j$$

holds for each $j=1, 2, \cdots, n$, and N has a total weight m, then the directed graph H' having $w(e)$ edges e_1 and $w(e')$ edges e_1' is a k-tournament with S as a score sequence, where $e=(v_j, v_q) \in K_n$, $e_1=(v_j, v_q) \in H'$, $e'=(v_q, v_j) \in K_n$, $e_1'=(v_q, v_j) \in H'$ for each $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$. Thus, to obtain a k-tournament, we obtain a weight w.

Suppose that S is separated to h sequences $U_1, U_2, \cdots, U_h (1 \leqq h \leqq n-1)$ now. Then, in the algorithm, variables cur mov and $F(j)$ and $L(j)$ for each $j=1, 2, \cdots, h$, are defined as follows:

(1) For each $j=1, 2, \cdots, h$, $U_j = (u_{F(j)}, u_{F(j)+1}, \cdots, u_{L(j)-1}, u_{L(j)})$ satisfies $0 \leqq u_{F(j)} \leqq u_{F(j)+1} \leqq \cdots \leqq u_{L(j)-1} \leqq u_{L(j)} \leqq k \cdot (L(j) - F(j))$ and

$$\sum_{i=F(j)}^{t} u_i \geqq k\cdot(t-F(j)+1)(t-F(j))/2$$

for each $t=F(j), F(j)+1, \cdots, L(j)$, with equality holding for $t=L(j)$,

(2) Assume that U_j is separated to two sequences $(u_{F(j)}, u_{F(j)+1}, \cdots, u_q)$ and $(u_{q+1}, u_{q+2}, \cdots, u_{L(j)})$ for some $j, 1\leqq j\leqq h$. Then $cur=j$, $mov:=h+1$(i.e., $U_{mov}:=(u_{q+1}, u_{q+2}, \cdots, u_{L(j)})$, $F(mov):=q+1$ and $L(mov):=L(j)$) and $U_j:=(u_{F(j)}, u_{F(j)+1}, \cdots, u_q)$ (i.e., $L(j):=q$).

Variables cur, $F(1)$ and $L(1)$ are initialized $cur=1$, $F(1)=1$ and $L(1)=n$.

Algorithm CkT-1.

Begin

01 **For** $j:=1$ **to** n **do** $u_j:=s_j$;
02 $cur:=1$; $mov:=1$; $num:=n$; $F(cur):=1$; $L(cur):=n$;
03 **While** $mov<num$ **do begin**
04 $g:=k\cdot(L(cur)-F(cur)+1)(L(cur)$
05 $-F(cur))/2-u_{L(cur)}$;
06 $a:=g-k\cdot(L(cur)-F(cur))(L(cur)$
$-F(cur)-1)/2$;
07 $b:=a$; $q:=L(cur)-1$;
08 **For** $j:=L(cur)-2$ **downto** $F(cur)$ **do begin**
09 $g:=g-u_{j+1}$;
10 $a:=g-k\cdot(j-F(cur)+1)(j-F(cur))/2$;
11 **If** $a<b$ **then begin** $b:=a$; $q:=j$ **end end**;
12 {{*Then* $b\geqq 0$ always holds.}}
13 **For** $j:=F(cur)$ **to** q **do begin**
14 $z:=min\{k, b\}$; $u_j:=u_j-z$;
15 $w((v_j, v_{L(cur)})):=z$;
16 $u_{L(cur)}:=u_{L(cur)}-k+z$;
17 $w((v_{L(cur)}, v_j)):=k-z$; $b:=b-z$ **end;**
18 **For** $j:=L(cur)-1$ **downto** $q+1$ **do begin**
19 **For** $h:=F(cur)$ **to** q **do begin**
20 **If** $h=F(cur)$ **then** $z:=min\{k, b\}$ **else** $z:=0$;
21 $u_h:=u_h-z$; $w((v_h, v_j)):=z$;
22 $u_j:=u_j-k+z$; $w((v_j, v_h)):=k-z$;
23 $b:=b-z$ **end end;**
24 **If** $q+1=L(cur)$ **then** $num:=num-1$
25 **else begin** $mov:=mov+1$;
26 $F(mov):=q+1$; $L(mov):=L(cur)$ **end;**
27 $L(cur):=q$;
28 **If** $L(cur)=F(cur)$ **then** $cur:=cur+1$ **end**

End.

From line 4 to 11, from line 13 to 17 and from line 18 to 23 of Algorithm *CkT*-1 correspond to Step1, Step6 and Step8 and 9 of Algorithm *kT*-2 respectively. While-loop repeats just $n-1$ times, and then we have $u_1=u_2=\cdots=u_n=0$. Hence Algorithm *CkT*-1 correctly constructs a k-tournament T with S as a score sequence. Since, from line 4 to 11 takes $O(n)$ time at once, and total edge-addition takes $O(n^2)$ time, Algorithm *CkT*-1 takes $O(n^2)$ time.

4. Second Approach

In this section, we consider another algorithm for constructing a k-tournament with $S=(s_1, s_2, \cdots, s_n)$ as the score sequence based on previous section. We can assume without loss of generality that $n\geqq 2$ and obtain the following proposition.

Proposition 4.1. 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 $U_1=(u_1, u_2, \cdots, u_q)$ and $U_2=(u_{q+1}, u_{q+2}, \cdots, u_n)$ be defined from S by the following algorithm, where $1\leqq q\leqq n-1$. Then S is a score sequence of a k-tournament if and only if U_1 and U_2 are score sequences of k-tournaments. Furthermore, $0\leqq u_1\leqq u_2\leqq\cdots\leqq u_q\leqq k(q-1)$ and $0\leqq u_{q+1}\leqq u_{q+2}\leqq\cdots\leqq u_n\leqq k(n-q-1)$.

Algorithm *kT*-3.

Step1: $q:=max\{t\mid min\{\sum_{j=1}^{t} s_j-kt(t-1)/2\}, 1\leqq t\leqq n-1\}$
and $b:=\sum_{j=1}^{t} s_j-kq(q-1)/2$.
{{i.e., $b<\sum_{j=1}^{t} s_j-kt(t-1)/2$
for each $t=q+1, q+2, \cdots, n-1$.}}

Step2: **If** $b<0$ **then** $b':=0$ **else** $b':=b$.

Step3: $u_1:=s_1-b'$ and $u_j=s_j$ for each $j=2, 3, \cdots, q$.

Step4: $u_n := s_n + b' - kq$ and $u_j := s_j - kq$ for each $j = q+1, q+2, \cdots, n-1$.

Note. Our object is to separate the sequence S to two sequences U_1 containing u_1 and U_2 containing u_n, by adding edges in $\{(v_j, v_h)$ and $(v_h, v_j) | j = 1, 2, \cdots, q, h = q+1, q+2, \cdots, n\}$ such that $|(v_j, v_h)| \geqq 0$, $|(v_h, v_j)| \geqq 0$ and $|(v_j, v_h)| + |(v_h, v_j)| = k$ hold and

$$f = \sum_{j=1}^{q} \sum_{h=q+1}^{n} |(v_j, v_h)| \geqq 0$$

is minimum. (In fact, we first add edges in $\{(v_h, v_j)\}$ such that $|(v_h, v_j)| = k$, and next we reverse the directions of edges (v_n, v_1) and, (v_n, v_j) and (v_j, v_1) for each $j = 2, 3, \cdots, n-1$, such that $|(v_n, v_j)| = |(v_j, v_1)|$ and

$$|(v_n, v_1)| + \sum_{j=2}^{n-1} |(v_n, v_j)| + \sum_{j=2}^{n-1} |(v_j, v_1)| = 2f.)$$

Furthermore, if S is a score sequence of a k-tournament then $b \geqq 0$ always holds in Step2 of Algorithm kT-3 by Proposition 4.1.

Proof. The necessity can be obtained as follows. Since

$$\sum_{j=1}^{1} s_j - k \cdot 1(1-1)/2 = s_1 \geqq b'$$

and

$$\sum_{j=1}^{n-1} s_j - k(n-1)(n-2)/2 = k(n-1) - s_n \geqq b',$$

we have $u_1 \geqq 0$ and $u_n \leqq k(n-q-1)$. Since

$$\sum_{j=1}^{t} s_j - \sum_{j=1}^{t} u_j = b' \geqq 0$$

for each $t = 1, 2, \cdots, q$, with equality holding for $t = q$,

$$\sum_{j=q+1}^{t} (s_j - kq) - \sum_{j=q+1}^{t} u_j = 0$$

for each $t = q+1, q+2, \cdots, n-1$, and

$$\sum_{j=q+1}^{n} u_j - \sum_{j=q+1}^{n} (s_j - kq) = b' \geqq 0,$$

we have

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

for each $t = 1, 2, \cdots, q$, with equality holding for $t = q$, and

$$\sum_{j=q+1}^{t} u_j \geqq k(t-q)(t-q-1)/2$$

for each $t = q+1, q+2, \cdots, n$, with equality holidg for $t = n$. Furthermore, it is clear that $0 \leqq u_1 \leqq u_2 \leqq \cdots \leqq u_q \leqq k(q-1)$ and $0 \leqq u_{q+1} \leqq u_{q+2} \leqq \cdots \leqq u_n \leqq k(n-q-1)$. Thus $U_1 = (u_1, u_2, \cdots, u_q)$ and $U_2 = (u_{q+1}, u_{q+2}, \cdots, u_n)$ are score sequences of k-tournaments by Proposition 2.4.

The sufficiency can be obtained as follows. Let H_1 and H_2 be k-tournaments with U_1 and U_2 as the score sequences respectively. Then $0 \leqq |(v_j, v_1)| \leqq k$ for each $j = 2, 3, \cdots, q$, and

$$\sum_{j=2}^{q} |(v_j, v_1)| = k(q-1) - u_1$$

in H_1 and $0 \leqq |(v_n, v_j)| \leqq k$ for each $j = q+1, q+2, \cdots, n-1$, and

$$\sum_{j=q+1}^{n-1} |(v_n, v_j)| = u_n$$

in H_2. Let T' be a k-tournament obtained from H_1 and H_2 by adding edges in $\{(v_h, v_j) | j = 1, 2, \cdots, q, h = q+1, q+2, \cdots, n\}$ such that $|(v_h, v_j)| = k$. Then T' has $S' = (s_1 - b', s_2, s_3, \cdots, s_{n-1}, s_n + b')$ as a score sequence. Since $|(v_n, v_1)| = k$, $|(v_n, v_j)| = k$ for each $j = 2, 3, \cdots, q$, and $|(v_j, v_1)| = k$ for each $j = q+1, q+2, \cdots, n-1$, in T' and $k + (k(q-1) - u_1) + u_n = 2b' + (s_n - s_1) \geqq 2b'$, we can obtain a k-tournament T with S as a score sequence from T' by the following algorithm. ■

Algorithm *RD*-1.

Step1: $R := b'$, $E_1 := \phi$ and $E_2 := \phi$.

Step2: $f_1 := min\{R, |(v_n, v_1)|\}$,
$E_1 := E_1 \cup \{f_1 \; edges \; (v_n, v_1)\}$,
$E_2 := E_2 \cup \{f_1 \; edges \; (v_1, v_n)\}$ and $R := R - f_1$.

Step3: **For** $j := 2$ **to** q **do** the fellowing (a) through (d).

(a) $f_j := min\{R, |(v_j, v_1)|\}$.

(b) $E_1 := E_1 \cup \{f_j \; edges \; (v_j, v_1)\}$ and f_j edges $(v_n, v_j)|$.

(c) $E_2 := E_2 \cup \{f_j \; edges \; (v_1, v_j)\}$ and f_j edges $(v_j, v_n)|$.

(d) $R := R - f_j$.

Step4: **For** $j := q+1$ **to** $n-1$ **do** the following (a) through (d).

(a) $f_j := min\{R, |(v_n, v_j)|\}$.

(b) $E_1 := E_1 \cup \{f_j \; edges \; (v_n, v_j)\}$ and f_j edges $(v_j, v_1)|$.

(c) $E_2 := E_2 \cup \{f_j \; edges \; (v_j, v_n)\}$ and f_j edges $(v_1, v_j)|$.

(d) $R := R - f_j$.

Step5: $T := (T' \cup E_2) - E_1$.

S is called an *ancestor* of U_1 and U_2, U_1 and U_2 are called *descendants* of S, and b' is called a *balance* of S. Algorithm kT-3 separates a sequence S to two sequences U_1 and U_2. Thus we can obtain $u_1=u_2=\cdots=u_n=0$ by using Algorithm kT-3 $n-1$ times, and can obtain the following iterative algorithm CkT-2 for constructing a k-tournament T with S as a score sequence, based on Proposition 2.4. In the following, we can assume without loss of generality that S is a score sequence of a k-tournament.

We define the complete directed graph K_n, the capacity function cap, the complete weighted directed graph $N=(K_n, cap)$ and the weight w by the same definition as in the previous section.

Suppose that S is separated to h sequences $U_1, U_2, \cdots, U_h$ $(1\leqq h\leqq 2n-1)$ now. Then, in the algorithm, variables cur, mov and $F(j)$, $L(j)$, $Anc(j)$ and $b(j)$ for each $j=1, 2, \cdots, h$, are defined as follows:

(1) For each $j=1, 2, \cdots, h$, $U_j=(u_{F(j)}, u_{F(j)+1}, \cdots, u_{L(j)-1}, u_{L(j)})$ satisfies $0\leqq u_{F(j)}=u_{F(j)+1}\leqq\cdots\leqq u_{L(j)-1}\leqq u_{L(j)}\leqq k\cdot(L(j)-F(j))$ and

$$\sum_{i=F(j)}^{t} u_i\geqq k\cdot(t-F(j)+1)(t-F(j))/2$$

for each $t=F(j), F(j)+1, \cdots, L(j)$, with equality holding for $t=L(j)$.

(2) Assume that $|U_j|\geqq 2$ and U_j is separated to two sequence $U'_{j1}=(u_{F(j)}, u_{F(j)+1}, \cdots, u_q)$ and $U'_{j2}=(u_{q+1}, u_{q+2}, \cdots, u_{L(j)})$ for some j, $1\leqq j\leqq h$. Then mov means descendants of U_j (i.e., $mov=h+1$ and $mov=h+2$) and $b(j)$ means a balance of U_j. Thus $cur=j$, $U_{h+1}=U'_{j1}$, $U_{h+2}=U'_{j2}$, $F(h+1)=F(cur)$, $L(h+1)=q$, $F(h+2)=q+1$, $L(h+2)=L(cur)$ and $Anc(h+1)=Anc(h+2)=cur$ hold.

(3) Assume that $|U_j|=1$ for some j, $1\leqq j\leqq h$. Then U_j does not have a descendant. Thus $cur=j$ and $b(j)=0$ hold.

Variables cur, $F(1)$, $L(1)$, $Anc(1)$ and mov are initialized $cur=1$, $F(1)=1$, $L(1)=n$, $Anc(1)=0$ and $mov=1$.

Algorithm CkT-2.

```
Begin
{{Initialization.}}
  For j:=1 to n do u_j:=s_j;
  For j:=1 to 2n-1 do begin
    Anc(j):=0; b(j):=0 end;
  cur:=1; mov:=1; F(cur):=1; L(cur):=n;
{{Separation.}}
  While mov<2n-1 do begin
    If F(cur)≠L(cur) then begin
      g:=k·(L(cur)-F(cur)+1)·
(L(cur)-F(cur))/2-u_L(cur);
      a:=g-k·(L(cur)-F(cur))·
(L(cur)-F(cur)-1)/2;
      b(cur):=a; q:=L(cur)-1;
      For j:=L(cur)-2 downto F(cur) do
      begin
        g:=g-u_{j+1};
        a:=g-k·(j-F(cur)+1)(j-F(cur))
        /2;
        If a<b(cur) then begin
          b(cur):=a; q:=j end end;
          {{Then b(cur)≧0 always holds.}}
      mov:=mov+1; F(mov):=F(cur):
      L(mov):=q;
      Anc(mov):=cur; mov:=mov+1;
      F(mov):=q+1;
      L(mov):=L(cur); Anc(mov):=cur;
      u_F(cur):=u_F(cur)-b(cur);
      u_L(cur):=u_L(cur)+b(cur);
      For j:=L(cur) downto q+1 do begin
        For h:=q downto F(cur) do begin
          u_j:=u_j-k; w((v_j, v_h)):=k;
          w((v_h, v_j)):=0 end end end;
    cur:=cur+1 end;
{{Unification.}}
  cur:=Anc(mov);
    {{Then mov=2n-1 holds.}}
  While cur≧1 do begin
     R:=b(cur); f:= min{R, w((v_L(cur),
    v_F(cur)))};
    w((v_L(cur), v_F(cur)))
      :=w((v_L(cur), v_F(cur)))-f;
    w((v_F(cur), v_L(cur)))
      :=w((v_F(cur), v_L(cur)))+f;
    R:=R-f;
    For j:=F(cur)+1 to L(mov-1) do
    begin
      f:=min{R, w((v_j, v_F(cur)))};
      w((v_L(cur), v_j)):=w((v_L(cur), v_j))-f;
```

```
43        w((v_j, v_F(cur))) := w((v_j, v_F(cur))) - f;
44        w((v_F(cur), v_j)) := w((v_F(cur), v_j)) + f;
45        w((v_j, v_L(cur))) := w((v_j, v_L(cur))) + f;
46        R := R - f end;
47      For j := F(mov) to L(cur) - 1 do begin
48        f := min{R, w((v_L(cur), v_j))};
49        w((v_L(cur), v_j)) := w((v_L(cur), v_j)) - f;
50        w((v_j, v_F(cur))) := w((v_j, v_F(cur))) - f;
51        w((v_F(cur), v_j)) := w((v_F(cur), v_j)) + f;
52        w((v_j, v_L(cur))) := w((v_j, v_L(cur))) + f;
53        R := R - f end;
54      mov := mov - 2; cur := Anc(mov) end
    End.
```

The relation between S and $\{U_1, U_2\}$ can be represented by a binary tree with n leaves. Then the tree has $2n-1$ vertices. Thus the sizes of F, of L, of Anc and of b are $2n-1$ respectively. "Separation" and "Unification" correspond to Algorithm kT-3 and RD-1 respectively. From line 9 to 19, and from line 25 to 28 correspond to Step1 and Step3 and 4 of Algorithm kT-3 respectively. From line 34 to 39, from line 40 to 46, and from line 47 to 53 correspond to Step1, 2 and 5, Step3 and 5, and Step4 and 5 of Algorithm RD-1 respectively. While-loops repeat just $n-1$ times respectively, and then we have $u_1=u_2=\cdots=u_n=0$ and a k-tournament with S as a score sequence. Since, from line 9 to 19 takes at most $O(n)$ time at once, and the total edge-addition takes $O(n^2)$ time, "Separation" takes $O(n^2)$ time. Since, from line 34 to 53 takes at most $O(n)$ time at once, "Unification" takes at most $O(n^2)$ time. Thus Algorithm CkT-2 takes $O(n^2)$ time.

5. Concluding Remarks

By the argument above, we can obtain 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 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.

We have given two optimal algorithms for constructing a k-tournament with $S=(s_1, s_2, \cdots, s_n)$ as a score sequence.

In the following, we obtain another characterization of a score sequence of an 1-tournament immediately based on Proposition 3.1 or 4.1 as follows.

Proposition 5.1. 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 $U_1=(u_1, u_2, \cdots, u_q)$ and $U_2=(u_{q+1}, u_{q+2}, \cdots, u_n)$ be defined from S by Algorithm CkT-1 or CkT-2 with $k=1$. Then S is a score sequence of an 1-tournament if and only if U_1 and U_2 are score sequeences of 1-tournaments. Furthermore, $0 \leqq u_1 \leqq u_2 \leqq \cdots \leqq u_q \leqq q-1$ and $0 \leqq u_{q+1} \leqq u_{q+2} \leqq \cdots \leqq u_n \leqq n-q-1$.

Of course, an 1-tournament with S as a score sequence can be constructed in $O(n^2)$ time by using the algorithm CkT-1 or CkT-2 with $k=1$.

Finally, we will consider other approaches of the score sequence pair problem of a bipartite k-tournament as our further investigations.

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, pp264-274, 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, pp496-506, 1962.

[7] Hakimi, S.L. and Moser, L., *The Theory of Round Robin Tournaments*, Amer. Math. Monthly 73, pp231-246, 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, pp477-480, 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, pp143-148, 1953.

[11] Menon, V.V., *On the Existence of Trees with Given Degrees,* Sankhyã, Series A, 26(1964), pp63-68.

[12] Takahashi, M., *A Consideration of a Score Sequence Problem of a Tournament*, Research Bulletin of Fukuoka Institute of Technology, Vol. 28. No.2, pp221-227, 1996.

[13] Takahashi, M., Imai, K. and Asano, T., *Graphical Degree Sequence Problems,* IEICE Trans. Fundamentals., Vol. E77-A. No.3 (March), pp546-552, 1994.