k-Edge-Connected Multigraphical Degree Sequence Problem

福岡工業大学研究論集 Res. Bull. Fukuoka Inst. Tech., Vol. 27 No. 2 (1995) 165—173

k-Edge-Connected Multigraphical Degree Sequence Problem

Masaya TAKAHASHI (Department of Electronics and Information Science, Fukuoka J. College of Technology)

Summary

A sequence of nonnegative integers $S=(s_1, s_2, \cdots, s_n)$ is *k-edge-connected multigraphical* if there is a k-edge-connected multigraph with vertices $v_1, v_2, \cdots, v_n$ such that deg $(v_j)=s_j$ for each $j=1, 2, \cdots, n$. The *k-edge-connected multigraphical degree sequence problem* is: Given a sequence of nonnegative integers, determine whether it is k-edge-connected multigraphical or not, where $k \geqq 1$. In this paper, I consider undirected version and directed version, and give a linear time algorithm respectively, for the both versions.

Key words: *nonnegative integers, k-edge-connected, multigraph, multidigraph and linear time algorithm*

1. Introduction

A sequence of nonnegative integers $S=(s_1, s_2, \cdots, s_n)$ is *graphical* if there is a graph with vertices $v_1, v_2, \cdots, v_n$ such that deg $(v_j)=s_j$ for each $j=1, 2, \cdots, n$ (deg (v_j) is the degree of v_j). The graphical degree sequence problem is: Given a sequence of nonnegative integers, determine whether it is graphical or not. The graphical degree sequence problem was first considered by Havel [10] and then considered by Erdös and Gallai [1] and Hakimi [4]. (There can be also found in standard books of graph theory [3, 5].)

Many variations can be considered. For example, if I admit multigraphs, then the multigraphical version is obtained. I recently studied variations described below [7, 8]. A set of sequences of nonnegative integers $\{S_1, S_2, \cdots, S_k\}$ with $S_j=(s_{j1}, s_{j2}, \cdots, s_{jn_j})$ is *k-partite graphical* (*k-partite multigraphical*) if there is a k-partite graph (k-partite multigraph) of k independent vertex sets $\{V_1, V_2, \cdots, V_k\}$ with $V_j=\{v_{j1}, v_{j2}, \cdots, v_{jn_j}\}$ such that deg $(v_{jq})=s_{jq}$ for each $j=1, 2, \cdots, k$ and $q=1, 2, \cdots, n_j$. The k-partite graphical (multigraphical) degree sequence problem is defined as follows: Given a set of sequences of nonnegative integers, determine whether it is k-partite graphical (multigraphical) or not. For these problems, Takahashi proposed characterizations leading to efficient algorithms [7]. However, his presentations were a little complicated and contained some holes. I also considered variations described below [8]:

(1) graphical degree sequence problem,
(2) multigraphical degree sequence problem,
(3) *bipartite graphical* degree sequence problem,
(4) *bipartite multigraphical* degree sequence problem,
(5) *k-partite multigraphical* degree sequence problem.

I presented efficient algorithms for (1) through (5) described above including $O(n)$ time

平成6年10月18日受付

algorithms to determine whether a (set of) nonnegative integer(s) is so or not and $O(r)$ time algorithms to construct such a graph having it as a (set of) degree sequence(s) if it is so, where

$$r=\begin{cases} n & \text{for (2) and (4) described above,} \\ m & \text{for (1) and (3) described above,} \\ n+k\log k & \text{for (5) described above,} \end{cases}$$

and ($m=\sum_{j=1}^{n}s_j/2$ or $m=\sum_{j=1}^{k}\sum_{q=1}^{n_j}s_{jq}/2$) [8].

The problems stated above are all about undirected graphs. Directed versions can also be defined [9]. For example, a pair of nonnegative integer sequences $\{S^+, S^-\}$ with $S^+=(s_1^+, s_2^+, \cdots, s_n^+)$ and $S^-=(s_1^-, s_2^-, \cdots, s_n^-)$ is *digraphical* (*multidigraphical*) if there is a directed graph (directed multigraph) with vertices $v_1, v_2, \cdots, v_n$ such that $\deg^+(v_j)=s_j^+$ and $\deg^-(v_j)=s_j^-$ for each $j=1, 2, \cdots, n$ ($\deg^+(v_j)$ and $\deg^-(v_j)$ are the outdegree and indegree of v_j respectively). The digraphical (multidigraphical) degree sequence problem is defined similarly: Given a pair of nonnegative integer sequences, determine whether it is digraphical (multidigraphical) or not. I also considered them and presented efficient algorithms including $O(n)$ time algorithms to determine whether a pair of nonnegative integer sequences $\{S^+, S^-\}$ is so or not and $O(r)$ time algorithms to construct such a directed graph having $\{S^+, S^-\}$ as a pair of degree sequences if $\{S^+, S^-\}$ is so, where

$$r=\begin{cases} n & \text{for multidigraphical version,} \\ m & \text{for digraphical version,} \end{cases}$$

and $m=\sum_{j=1}^{n}s_j^+ \ (=\sum_{j=1}^{n}s_j^-)$.

In this paper, I consider the *k-edge-connected multigraphical* degree sequence problem, and present an efficient algorithm including an $O(n)$ time algorithm to determine whether a sequence of nonnegative integers $S=(s_1, s_2, \cdots, s_n)$ is k-edge-connected multigraphical or not and an $O(n)$ time algorithm to construct a k-edge-connected multigraph having S as a degree sequence if S is k-edge-connected multigraphical.

Furthermore, I consider the *k-edge-connected multidigraphical* degree sequence problem, and present an efficient algorithm including an $O(n)$ time algorithm to determine whether a pair of sequences of nonnegative integers $\{S^+, S^-\}$ with $S^+=(s_1^+, s_2^+, \cdots, s_n^+)$ and $S^-=(s_1^-, s_2^-, \cdots, s_n^-)$ is k-edge-connected multidigraphical or not and an $O(n)$ time algorithm to construct a k-edge-connected directed multigraph having $\{S^+, S^-\}$ as a pair of degree sequences if $\{S^+, S^-\}$ is k-edge-connected multidigraphical.

2. k-Edge-Connected Multigraphical Degree Sequence Problem

In this section, I consider the k-edge-connected multigraphical (kECM-graphical, for short) degree sequence problem: Given a sequence of nonnegative integers $S=(s_1, s_2, \cdots, s_n)$, determine whether it is kECM-graphical (that is, there is a k-edge-connected multigraph (kEC-multigraph, for short) with vertices $v_1, v_2, \cdots, v_n$ such that $\deg(v_j)=s_j$ for each $j=1, 2, \cdots, n$).

I assume that $\sum_{j=1}^{n}s_j$ is even, because otherwise S is not kECM-graphical. I can assume $s_1 \geqq s_2 \geqq \cdots \geqq s_n$ without loss of generality. If $n=2$ then $s_1=s_2$ and the kEC-multigraph can be constructed easily. Thus I assume throughout this section that $\sum_{j=1}^{n}s_j$ is even, $n \geqq 3$ and $s_1 \geqq s_2 \geqq \cdots \geqq s_n$.

I first consider the $k \geqq 2$ case.

Then the following proposition can be obtained easily.

Proposition 2. 1. Let $S=(s_1, s_2, \cdots, s_n)$ be a sequence of nonnegative integers such that $s_1 \geqq s_2 \geqq \cdots \geqq s_n$ and $\sum_{j=1}^{n}s_j$ is even. Then S is kECM-graphical if and only if $s_1 \leqq \sum_{j=1}^{n}s_j$ and $s_n=k$.

In the following, I show the proof of the proposition.

Necessity is almost trivial. If S is kECM-graphical then it is multigraphical. Thus $s_1 \leqq$

$\sum_{j=2}^{n} s_j$ holds [8]. Furthermore, if $s_j < k$ for some $j=1, 2, \cdots, n$, then S is not kECM-graphical clearly. Thus $s_j \geqq k$ holds for each $j=1, 2, \cdots, n$. (Especially, $s_n = k$.)

The sufficiency can be obtained as follows. Let K_n be the complete graph with vertex set $V=\{v_1, v_2, \cdots, v_n\}$ and let $N=(K_n, \text{cap})$ be the weighted graph defined by the capacity function cap $(e) = \infty$ for $e=(v_j, v_q)$, $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$. Since S is kECM-graphical, N has a weight of value $m=\sum_{j=1}^{n} s_j/2$. For a weight w of N of value m, I creat $w(e)$ copies of an edge $e=(v_j, v_q)$, for each $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$ with $w(e) \neq 0$. Then the kEC-multigraph H_w obtained in this way has S as a sequence of nonnegative integers. To obtain a weight w of value m (maximum weight), I first obtain weights r_1 and r_2 of N. Then r_1 and r_2 can be obtained as follows (I assume $r_1(e)=0$ and $r_2(e)=0$ for $e \in E(K_n)$ at the beginning, where $E(K_n)$ is an edge set of K_n).

Algorithm A.

Begin

1. $p := k/2$; $L_1 := \Phi$; $L_2 := \Phi$;
   {$\lfloor t \rfloor$ is the greatest integer $\leqq t$. L_1 and L_2 are edge sets and represented by linked lists.}
   If (k and n are odd) and ($s_1 = s_2$) then
   $y := \max\{ j \mid s_j = s_1 \}$ else $y := 1$;
2. For $j:=1$ to $n-1$ do begin
   $r_1(e) := r_1(e) + p$ and $r_2(e) := r_1(e)$
   for $e=(v_j, v_{j+1})$;
   If $j \neq y$ and $j+1 \neq y$ then
   insert an edge $e=(v_j, v_{j+1})$ into L_1 end;
   $r_1(e) := r_1(e) + p$ and $r_2(e) := r_1(e)$
   for $e=(v_n, v_1)$;
   If $y \neq 1$ and $y \neq n$ then
   insert an edge $e=(v_n, v_1)$ into L_1 ;
   For $j:=1$ to n do $s_j' := s_j - 2p$;
   {since $s_j \geqq k$ for each $j=1, 2, \cdots, n$.}
3. If k is odd then begin
   $x := n/2$; { t is the smallest integer $\geqq t$.}
   For $j:=1$ to $x-1$ do begin
   $r_2(e) := r_2(e) + 1$ for $e=(v_j, v_{j+x})$;
   If $y \neq j$ and $y + j + x$ then
   insert an edge $e=(v_j, v_{j+x})$ into L_2 end;
   $r_2(e) := r_2(e) + 1$ for $e=(v_1, v_{1+x})$;
   If $y \neq 1$ and $y \neq 1+x$ then
   insert an edge $e=(v_1, v_{1+x})$ into L_2 ;
   If n is odd then begin
   $r_2(e) := r_2(e) + 1$ for $e=(v_x, v_1)$;
   If $y \neq 1$ and $y \neq x$ then
   insert an edge $e=(v_1, v_x)$ into L_2 end;
   For $j:=2$ to n do $s_j' := s_j' - 1$;
   If n is odd then $s_1' := s_1' - 2$
   else $s_1' := s_1' - 1$ end;
4. $v_j' := v_j$ for each $j=1, 2, \cdots, n$;
   If (k and n are odd) and ($s_1 = s_2$) then begin
   swap $\{s_1', s_y'\}$; swap $\{v_1', v_y'\}$ end

End.

Let G_1 be a multigraph obtained from N as follows: N has a weight $r_2(e)$ for each $e=(v_j, v_q)$ if and only if G_1 has $r_2(e)$ edges (v_j, v_q) for each $j \neq q$, $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$. Then G_1 is k-edge-connected.

Next I obtain a maximal weight g of N as follows. Let $A=s_1'$ and $B=\sum_{j=2}^{n} s_j'$. Then g is obtained as follows. (I assume $g(e)=r_2(e)$ for $e \in E(K_n)$ at the beginning. Furthermore, I use the multigraph construction algorithm [8] if $A \leqq B$.)

Algorithm B.

Begin

1. $L_3 := \Phi$;
   {L_3 is an edge set and represented by a linked list.}
2. If $A > B$ then begin
   $s_1'' := B$;
   For $j:=2$ to n do begin
   $g(e) := g(e) + s_j'$ for each $e=(v_1', v_j')$;
   $s_1'' := s_1'' - s_j'$; $s_j' := 0$ end end
3. else begin

For $j := n$ downto 4 do begin
$\triangle := s_1' - s_2'$;
If $\triangle \geqq s_j'$ then begin
$s_1' := s_1' - s_j'$; $s_j' := 0$;
$g(e) := g(e) + s_j'$ for each $e = (v_1', v_j')$ end
else begin
$s_1' := s_1' - \triangle$: $s_j' := s_j' - \triangle$;
$g(e) := g(e) + \triangle$ for each $e = (v_1', v_j')$;
$s_{j-1}' := s_{j-1}' - s_j'$; $s_j' := 0$;
$g(e) := g(e) + s_j'$ for each $e = (v_{j-1}', v_j')$;
insert an edge $e = (v_{j-1}', v_j')$ into L_3 end
end;
$g(e) := g(e) + (s_1' + s_2' - s_3')/2$ for $e = (v_1', v_2')$;
$g(e) := g(e) + (s_2' + s_3' - s_1')/2$ for $e = (v_2', v_3')$;
insert an edge $e = (v_2', v_3')$ into L_3;
$g(e) := g(e) + (s_3' + s_1' - s_2')/2$ for $e = (v_3', v_1')$;
$s_1' := 0$; $s_2' := 0$; $s_3' := 0$ end
End.

Let G_2 be a multigraph obtained from N as follows: N has a weight $g(e)$ for each $e = (v_j', v_q')$ if and only if G_2 has $g(e)$ edges (v_j', v_q') for each $j \neq q$, $j = 1, 2, \cdots, n-1$, $q = j+1, j+2, \cdots, n$. Then G_2 is k-edge-connected clearly.

If $A \leqq B$ then g is a maximum weight of N of value m. Otherwise, I can obtain the following lemma.

Lemma 2. 2. Let $b = A - B$ and $c = 2\sum_{j=2}^{n} \sum_{q=j+1}^{n} ((v_j', v_q'))$. Then $b \leqq c$ holds and, b and c are even.

Proof. Assume that k is even. Then $b = A - B = (s_1 - k) - \sum_{j=2}^{n}(s_j - k) = s_1 - \sum_{j=2}^{n} s_j + k(n-2)$ and $c = kn - 2k = k(n-2)$ hold and c is even. Assume that k is odd and n is even. Then $b = s_1 - \sum_{j=2}^{n} s_j + k(n-2)$ and $c = k(n-2)$ hold and c is even similarly. Assume that k and n are odd. Then $b = A - B = (s_1 - (k+1)) - \sum_{j=2}^{n}(s_j - k) = s_1 - \sum_{j=2}^{n} s_j + k(n-2) - 1$ and $c = (k+1) + k(n-1) - 2(k+1) = k(n-2) - 1$ hold and c is even. Thus $b - c = s_1 - \sum_{j=2}^{n} s_j \leqq 0$ holds since $s_1 \leqq \sum_{j=2}^{n} s_j$, and c is even. Then b is even clearly since $A + B$ is even and $b = A - B$. □

Hence, if $A > B$ then I can obtain the maximum weight w of N of value m as follows (I assume $w(e) = g(e)$ for $e \in E(K_n)$ at the beginning).

Algorithm C.
Begin
1. $b := A - B$;
2. For $j := 1$ to 3 do begin
while L_j is not empty do begin
delete an edge $e = (v_r', v_q')$ from L_j;
$\{r \neq q, j = 2, 3, \cdots, n, q = r+1, r+2, \cdots, n.\}$
If $j = 1$ then $t(e) := r_1(e)$
else if $j = 2$ then $t(e) := r_2(e) - r_1(e)$
else $t(e) := g(e) - r_2(e)$;
If $b/2 \leqq t(e)$ then $z := b/2$ else $z := t(e)$;
$w(e) := g(e) + z$ for $e = (v_1', v_r')$;
$w(e) := g(e) + z$ for $e = (v_1', v_q')$;
$w(e) := g(e) - z$ for $e = (v_r', v_q')$;
$b := b - 2z$ end;
If $b = 0$ then *go* to Step 3 end end;
3. halt
End.

Let G be a multigraph obtained from N as follows: N has a weight $w(e)$ for each $e = (v_j', v_q')$ if and only if G has $w(e)$ edges (v_j', v_q') for each $j \neq q$, $j = 1, 2, \cdots, n-1$, $q = j+1, j+2, \cdots, n$. G_2 has $g(e)$ edges for each $e = (v_j', v_q')$, $j \neq q$, $j = 1, 2, \cdots, n-1$, $q = j+1, j+2, \cdots, n$, and therefore, G has at least $g(e)$ edge-disjoit v_j'-v_q' paths by the behavior of Algorithm C. Hence G is k-edge-connected. Hence, I can obtain the kEC-multigraph H_w with S as a degree sequence of nonnegative integers.

This completes the proof of Proposition 2. 1.

Based on Proposition 2. 1, I can determine in

$O(n)$ time easily whether S is kECM-graphical or not.

For the kEC-multigraph construction, it is clear that Algorithm A and C described above can perform in $O(n)$ time [8] respectively, and Algorithm B described above can perform in $O(n)$ time.

Next, I consider the $k=1$ case.

Then the following proposition can be obtained easily.

Proposition 2. 3. Let $S=(s_1, s_2, \cdots, s_n)$ be a sequence of nonnegative integers such that $s_1 \geqq s_2 \geqq \cdots \geqq s_n$ and $\sum_{j=1}^{n} s_j$ is even. Then S is connected multigraphical if and only if $s_1 \leqq \sum_{j=2}^{n} s_j$, $\sum_{j=1}^{n} s_j \geqq 2(n-1)$ and $s_n=1$.

In the following, I show the proof of the proposition.

Necessity is almost trivial. By the similar discussion of the previous case, $s_1 \leqq \sum_{j=2}^{n} s_j$ and $s_j \geqq 1$ for each $j=1, 2, \cdots, n$. (Especially, $s_n=1$.) Furthermore, if $\sum_{j=1}^{n} s_j < 2(n-1)$ then S is not connected multigraphical. Thus $\sum_{j=1}^{n} s_j \geqq 2(n-1)$ holds.

The sufficiency can be obtained by the similar discussion of the previous case as follows. To obtain a weight w of value $m=\sum_{j=1}^{n} s_j/2$ (maximum weight), I first obtain weights r_1 and r_2 of N. Then r_1 and r_2 can be obtained as follows (I assume $r_1(e)=0$ and $r_2(e)=0$ for $e \in E(K_n)$ at the beginning).

Algorithm D.

Begin

1. $x := \min\{j \mid s_j=1\}$; $L_1 := \Phi$; $L_2 := \Phi$;
   $z := n$;
   {L_1 and L_2 are edge sets and represented by linked lists.}
2. For $j:=1$ to $x-1$ do Edgad 1 $(j, j+1)$;
   For $j:=x-1$ downto 1 do begin
   $y:=s'_j$;
   For $q:=y$ downto 1 do begin
   Edgad 1 (j, z) ; $z:=z-1$;
   If $z=x$ then *go to* 3 end end;
3. $v'_j:=v_j$ for each $j=1, 2, \cdots, n$

End

Procedure Edgad 1 (r, t).

Begin

1. $r_2(e):=r_2(e)+1$ for $e=(v_r, v_t)$;
2. $s'_r:=s_r-1$; $s'_t=s_t-1$;
3. If $r \neq 1$ then insert an edge $e=(v_r, v_t)$ into L_2

End.

Then $s'_1 \geqq s'_2 \geqq \cdots \geqq s'_n$ holds clearly.

Let G_1 be a multigraph obtained from N as follows: N has a weight $r(e)$ for each $e=(v'_j, v'_q)$ if and only if G_1 has $r(e)$ edges (v'_j, v'_q) for each $j \neq q$, $j=1, 2, \cdots, n-1$, $q=j+1, j+2, \cdots, n$. Then G_1 is connected.

Obviously I can obtain a maximum weight w of N by the same discussion of the previous case, i.e., I can use Algorithm B and C.

Hence, I can obtain the connected multigraph H_w with S as a degree sequence of nonnegative integers.

This completes the proof of Proposition 2. 3.

Based on Proposition 2. 3, I can determine in $O(n)$ time easily whether S is connected multigraphical or not.

For the connected multigraph construction, it is clear that Algorithm D described above can perform in $O(n)$ time.

Thus, by above discussions, I can obtain the following theorem.

Theorem 1. For a sequence of nonnegative integers $S=(s_1, s_2, \cdots, s_n)$, it can be determined in $O(n)$ time whether S is kECM-graphical and, if so, a kEC-multigraph G with S as a degree sequence of nonnegative integers can be constructed in $O(n)$ time.

3. k-Edge-Connected Multidigraphical Degree Sequence Problem

In this section, I consider the k-edge-connected multidigraphical (kECM-digraphical, for short) degree sequence problem: Given a pair of nonnegative integer sequences $\{S^+, S^-\}$ with $S^+=(s_1^+, s_2^+, \cdots, s_n^+)$ and $S^-=(s_1^-, s_2^-, \cdots, s_n^-)$, determine whether it is kECM-digraphical (that is, there is a k-edge-connected directed multigraph (kEC-directed multigraph, for short) with vertices $v_1, v_2, \cdots, v_n$ such that $\deg^+(v_j)=s_j^+$ and $\deg^-(v_j)=s_j^-$ for each $j=1, 2, \cdots, n$, where $k \geqq 1$.

The following proposition can be obtained easily.

Proposition 3. 1. Let $\{S^+, S^-\}$ be a pair of nonnegative integer sequences with $S^+=(s_1^+, s_2^+, \cdots, s_n^+)$ and $S^-=(s_1^-, s_2^-, \cdots, s_n^-)$. Then $\{S^+, S^-\}$ is kECM-digraphical if and only if the following conditions $(C1)$ through $(C3)$ hold:

$(C1)$ $\sum_{q=1}^n s_q^+ = \sum_{q=1}^n s_q^-$,

$(C2)$ $s_j^+ + s_j^- \leqq \sum_{q=1}^n s_q^+ =$ for each $j=1, 2, \cdots, n$,

$(C3)$ $s_j^+ \geqq k$ and $s_j^- \geqq k$ for each $j=1, 2, \cdots, n$.

In the following, I show the proof of the proposition.

Necessity is almost trivial. If $\{S^+, S^-\}$ is kECM-digraphical then it is multidigraphical. Thus the conditions $(C1)$ and $(C2)$ described above hold[8]. Furthermore, if $s_j^+ < k$ or $s_j^- < k$ for some $j=1, 2, \cdots, n$, then $\{S^+, S^-\}$ is not kECM-digraphical clearly. Thus condition $(C3)$ described above also holds.

The sufficiency can be obtained as follows. Let $K_{n,n}$ be the complete bipartite graph with two independent vertex sets $V^+=\{v_1^+, v_2^+, \cdots, v_n^+\}$ and $V^-=\{v_1^-, v_2^-, \cdots, v_n^-\}$ and let F be the graph obtained from $K_{n,n}$ by adding two vertices s, t and the directed edges in $\{(s, v_j^+) \mid j=1, 2, \cdots, n\} \cup \{(v_j^-, t) \mid j=1, 2, \cdots, n\}$ and deleting all edges in $\{(v_j^+, v_j^-) \mid j=1, 2, \cdots, n\}$. I consider all edges of $K_{n,n}$ are directed from V^+ to V^-. Let $N=(F, \text{cap}, s, t)$ be the network defined by the capacity function

$$\text{cap}(e)=\begin{cases} s_j^+ & (e=(s, v_j^+), j=1, 2, \cdots, n) \\ \infty & (e=(v_j^+, v_q^-), j \neq q, j=1, 2, \cdots, n, \\ & \quad q=1, 2, \cdots, n) \\ s_j^- & (e=(v_j^-, t), j=1, 2, \cdots, n). \end{cases}$$

Since $\{S^+, S^-\}$ is kECM-digraphical, N has a flow of value $m=\sum_{q=1}^n s_q^+ = \sum_{q=1}^n s_q^-$. For a flow f of N of value m, I creat $f(e)$ copies of a directed edge $e=(v_j^+, v_q^-)$, for each $j=1, 2, \cdots, n$, $q=1, 2, \cdots, n$ with $f(e) \neq 0$. Then the kEC-directed multigraph H_f obtained in this way has $\{S^+, S^-\}$ as a pair of degree sequences. To obtain a flow f of value m (maximum flow), I first obtain a flow r of N. Then r can be obtained as follows (I assume $r(e) = 0$ for $e \in E(F)$ at the beginning, where E(F) is an edge set of F).

Algorithm E.

Begin

1. For $j:=2$ to n do begin
   $r(e):=r(e)+k$
   for each $e=(s, v_j^+), (v_j^+, v_{j-1}^-), v_{j-1}^-, t)$;
   $t_j^+:=s_j^+-k$; $t_{j-1}^-:=s_{j-1}^--k$ end;
2. $r(e):=r(e)+k$ for each $e=(s, v_1^+), (v_1^+, v_n^-), (v_n^-, t)$;
   $t_1^+:=s_1^+-k$; $t_n^-:=s_n^--k$
   {since $s_j^+ \geqq k$ and $s_j' \geqq k$ for each $j=1, 2, \cdots, n.$}

End.

Let G_1 be a directed multigraph obtained as follows: N has a flow $r(e)$ for each $e=(v_j^+, v_q^-)$ if and only if G_1 has $r(e)$ edges from v_j to v_q for each $j \neq q$, $j=1, 2, \cdots, n$, $q=1, 2, \cdots, n$. Then G_1 is kECM-digraphical clearly.

Next I obtain a maximal flow g of N by using Dinic's way[11] as follows. Let L^+ be doubly-linked list of $T^+=\{t_1^+, t_2^+, \cdots, t_n^+\}$ with one extra element $t_{n+1}^+=0$. L^+ is maintained to contain only $t_j^+\neq 0$ except $t_{n+1}^+=0$. ft^+ be the first element of L^+. ind is an index function of elements of L^+ (i. e. ind$(t_j^+)=j$). pre[x] and suc[x] denote the previous element and the next element of $x\in L^+$. Then g can be obtained as follows (I assume $g(e)=r(e)$ for $e\in E(F)$ at the beginning).

Algorithm F.

```
Begin
  For j:=1 to n do begin
    x:=ft⁺; q:=ind(x);
      while t_j⁻>0 and j≠n+1 do begin
        If q=j then begin x:=suc[x];
        q:=ind(x) end
        else begin
          p:=min{t_j⁻, t_q⁺};
          g(e):=g(e)+p
            for each e=(s, v_q⁺), (v_q⁺, v_j⁻), (v_j⁻,
            t);
        t_q⁺:=t_q⁺−p; t_j⁻:=t_j⁻−p;
        If t_q⁺=0 then begin
          x:=suc[x]; q:=ind(x);
          delete pre[x] from L⁺ end
        {else t_j⁻=0} end end end
End.
```

Let G_2 be a directed multigraph obtained as follows: N has a flow $g(e)$ for each $e=(v_j^+, v_q^-)$ if and only if G_2 has $g(e)$ edges from v_j to v_q for each $j\neq q, j=1, 2, \cdots, n, q=1, 2, \cdots, n$. Then G_2 is also kECM-digraphical clearly.

Let $N(g)$ be the residual capacity network of flow g of N. Edge set $E(N(g))$ of $N(g)$ is

$$E(N(g))=\{e\in E(F)\mid g(e)<\text{cap}(e)\}\cup\{e_{rev}=(u,w)\mid e=(w,u)\in E(F), g(e)>0\},$$

where e_{rev} is the reverse edge of e. Capacity function cap' of $N(g)$ satisfies

$$\text{cap}'(e)=\begin{cases}\text{cap}(e)-g(e) & (e\in\{e\in E(F)\mid g(e)<\text{cap}(e)\})\\ g(e_{rev}) & (e_{rev}\in\{e_{rev}\in E(F)\mid g(e_{rev})>0).\end{cases}$$

Note that there is at most one edge from s to V^+, because cap $(e)=\infty$ for each $e=(v_j^+, v_q^-)$ with $j\neq q$. Similarly, there is at most one edge from V^- to t. If there is no such edge then g is a maximum flow of value m. Otherwise, let (s, v_j^+) and (v_q^-, t) be such edges. Then $j=q$, because cap $(e)=\infty$ for each $e=(v_p^+, v_h^-)$ with $p\neq h$. Thus cap$(s, v_j^+)-g(s, v_j^+)=$cap $(v_j^-, t)-g(v_j^-, t)$. Then, since cap $(s, v_j^+)+$cap $(v_j^-, t)=s_j^++s_j^-\leqq\sum_{a=1}^{p}$ cap $(s, v_a^+)=\sum_{a=1}^{p}s_a^+$,

$$\sum_{e_{rev}\in N(g)} g(e_{rev})=\sum_{a=1}^{p}\text{cap}(s, v_a^+)-\text{cap}(v_j^-, t)-g(s, v_j^+)\geqq \text{cap}(s, v_j^+)-g(s, v_j^+)$$

always holds. Hence, by augmenting $\triangle=$ cap $(s, v_j^+)-g(s, v_j^+)$ flow along the augmenting paths $(s, v_j^+), (v_j^+, v_q^-), (v_q^-, v_h^+), (v_h^+, v_j^-), (v_j^-, t)$ of $N(g)$ of length 5 from s to t, I can obtain a maximum flow f of N, where (v_q^-, v_h^+) is the reverse edge of (v_h^+, v_q^-).

Let G be a directed multigraph obtained as follows: N has a flow $f(e)$ for each $e=(v_j^+, v_q^-)$ if and only if G has $f(e)$ edges from v_j to v_q for each $j\neq q, j=1, 2, \cdots, n, q=1, 2, \cdots, n$. For G, let U and W be any two independent vertex sets of V satisfying $V=U\cup W$. Let $E_G(U, W)=\{(u, w)\mid u\in U$ and $w\in W\}$ and $E_G(W, U)=\{(w, u)\mid u\in U$ and $w\in W\}$ be two edge sets of G. Similarly, I define $E_{G_2}(U, W)$ and $E_{G_2}(W, U)$ for G_2. For an augmenting path $P=(s, v_j^+), (v_j^+, v_q^-), (v_q^-, v_h^+), (v_h^+, v_j^-), (v_j^-, t)$ of $N(g)$, let $F(P)$ be a flow of P in $N(g)$. Since G_2 is k-edge-connected, $|E_{G_2}(U, W)|\geqq k$ and $|E_{G_2}(W, U)|\geqq k$. Then, for P,

$$f(e)=\begin{cases}g(e)+F(P) & \text{if } e=(v_h^+, v_j^-) \text{ or } e=(v_j^+, v_q^-),\\ g(e)-F(P) & \text{if } e=(v_h^+, v_q^-),\\ g(e) & \text{otherwise.}\end{cases}$$

If $v_j, v_q, v_h\in U$ or $v_j, v_q, v_h\in W$ then $|E_G(U, W)|\geqq k$ and $|E_G(W, U)|\geqq k$ for P clearly. I consider the following three cases:

(1) Assume $v_j, v_q\in U$ and $v_h\in W$. Then $|E_G(U, W)|=|E_{G_2}(U, W)|$ and $|E_G(W, U)|=|$

$E_{G_2}(W, U)|$ for P.

(2) Assume $v_j \in U$ and $v_q, v_h \in W$. Then $|E_G(U, W)| = |E_{G_2}(U, W)| + F(P)$ and $|E_G(W, U)| = |E_{G_2}(W, U)| + F(P)$ for P.

(3) Assume $v_j, v_h \in U$ and $v_q \in W$. Then $|E_G(U, W)| = |E_{G_2}(U, W)|$ and $|E_G(W, U)| = |E_{G_2}(W, U)|$ for P.

Repeating the argument above for all augmenting paths of $N(g)$, I can obtain that $|E_G(U, W)| \geqq k$ and $|E_G(W, U)| \geqq k$.

Thus, I can obtain the kECM-directed graph H_f with $\{S^+, S^-\}$ as a pair of degree sequences.

This completes the proof of Proposition 3. 1.

Based on Proposition 3. 1, I can determine in $O(n)$ time easily whether $\{S^+, S^-\}$ is kECM-digraphical or not.

For the kECM-directed graph construction, Algorithm E described above can perform in $O(n)$ time clearly, and Algorithm F and after it described above of the proof of the proposition can perform in $O(n)$ time.

Thus I can obtain the following theorem.

Theorem 2. For a pair of nonnegative integer sequences $S^+=(s_1^+, s_2^+, \cdots, s_n^+)$ and $S^-=(s_1^-, s_2^-, \cdots, s_n^-)$, it can be determined in $O(n)$ time whether $\{S^+, S^-\}$ is kECM-digraphical and, if so, a kECM-directed graph G with $\{S^+, S^-\}$ as a pair of degree sequences can be constructed in $O(n)$ time.

4. Concluding Remarks

In this paper, I have considered the k-edge-connected multigraphical degree sequence problem and the k-edge-connected multidigraphical one, and give linear time algorithms for them. Algorithms described here constructing such graphs are all concerned with representing a graph explicitly. If an implicitly represented graph is satisfactory, I can obtain faster O(n loglog n) time algorithms for constructing graphs, bipartite graphs and k-connected graphs (see [2]).

Takahashi [7] considered the k-partite graphical degree sequence problem and presented an $O(kn^2)$ time algorithm. However, I have found that there is a hole in his proof and his algorithm works only for 3-partite graph. Although this problem is polynomially solvable based on maximum matching algorithms[6]. I conjecture that there may be O(km) algorithms for k-partite digraphical degree sequence problem and k-partite graphical one.

Furthermore, I will consider the k-edge-connected digraphical degree sequence problem and present efficient algorithms for further investigation.

References

[1] P. Erdös and T. Gallai, *Graphs with prescribed degrees of vertices* (Hungarian), Mat. Lapok 11 (1960), pp264-274.

[2] T. Asano, *An O (n loglog n) time algorithm for constructing a graph of maximum connectivity with a prescribed degrees*, Technical Report, ISETR-93-02, Department of Information and System Engineering, Chuo University, 1993.

[3] M. Behzad, G. Chartrand and L.L-Foster, *Granphs and Digraphs*, Prindle, Weber and Schmidt, (1979).

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

[5] F. Harary, *Graph Theory*, Addison-Wesley, (1969).

[6] J. Edmonds and E.L. Johnson, *Matching: a well-solved class of integer linear programs, in Combinatorial Structures and Their Applications*, Gordon and Breach, 1970, pp 89-92.

[7] M. Takahashi, *An algorithm of constructing a k-partite graph from a k-partite graphical sequence set*, Technical Report TR92-AL-30-4, Information Processing Society of Japan, 1992.

[8] M. Takahashi, K. Imai and T. Asano, *Graphical degree sequence problems*, IEICE Trans.

Fundamentals., E77-A (No. 3, 1994), pp 546-552.

[9] Y.K. Chen, *Applied Graph Theory* (Second revised edition), North-Holland, (1976).

[10] V. Havel, *A remark on the existence of finite graphs* (Czech), Casopis Pest, Mat, 80 (1955), pp477-480.

[11] E.A. Dinic, *Algorithm for solution of a problem of maximum flow in a network with power estimation*, Soviet Mathematics Doklady 11 (1970), pp1277-1280.