アルゴリズム 28－3
（1992．7．17）

k組多重グラフ的次数列集合からk組多重グラフを構成するアルゴリズム

高橋 昌也 （愛知技術短期大学電子工学科）

$k \geqq 3$ なる任意の整定数 k について、グラフ G＝（V，E） がk組(多重)グラフであるとは、次の（1）～（3）が成り立つような頂点集合 V が存在することである。

（1） $V = V_1 \cup V_2 \cup \cdots \cup V_k$,

（2） 任意の2つの整数 h, j ($1 \leqq h \leqq k$, $1 \leqq j \leqq k$, $h \neq j$) について、$V_h \cap V_j = \Phi$、

（3） 任意の辺 $e=(u,v) \in E$ について、$u \in V_j$ ($1 \leqq j \leqq k$) ならば $v \notin V_j$ が成り立つ。

本稿では、k 個の次数列 $s_j : d_{j1}, d_{j2}, \cdots, d_{j,pj}$ ($1 \leqq j \leqq k$) と k 個の頂点集合 $V_j = \{v_{j1}, v_{j2}, \cdots, v_{j,pj}\}$ が与えられた時、次の（1）（2）を行う線形時間アルゴリズムを提案する。

（1） $S = (s_1, s_2, \cdots, s_k)$ がk組多重グラフ的次数列集合であるか否かを判定する。

（2） S がそうならば、$1 \leqq j \leqq k$, $1 \leqq q \leqq p_j$ なる j, q について、v_{jq} に入っている辺の数が d_{jq} であるようなk組多重グラフを構成する。

An Algorithm of Constructing a k-partite Multigraph from a k-partite Multigraphical Sequence Set

Masaya Takahashi (Department of Electronic Engineering, Aichi College of Technology)

For a given integer constant k which satisfies $k \geqq 3$, and a graph G＝（V，E）, G is called a k-partite (multi) graph if there is a set of vertices V such that the following (1) through (3) are satisfied： (1) $V = V_1 \cup V_2 \cup \cdots \cup V_k$,

(2) For any two integer h and j, $1 \leqq h \leqq k$, $1 \leqq j \leqq k$, $h \neq j$, $V_h \cap V_j = \Phi$ is satisfied, and

(3) For any edge $e=(u,v)$, if $u \in V_j$ then $v \notin V_j$ is satisfied, where $1 \leqq j \leqq k$.

In this paper, when k digree sequences $s_j : d_{j1}, d_{j2}, \cdots, d_{j,pj}$, for every j, $1 \leqq j \leqq k$, and k sets of vertices $V_j = \{v_{j1}, v_{j2}, \cdots, v_{j,pj}\}$, for every j, $1 \leqq j \leqq k$, are given, propose an algorithm satisfying the following (1) through (3)：

(1) Decide that whether a given non-negative integer sequence set $S = (s_1, s_2, \cdots, s_k)$ is a degree sequence set of a k-partite multigraph.

(2) If $S = (s_1, s_2, \cdots, s_k)$ is a degree sequence set of a k-partite multigraph then construct a k-partite multigraph $G = (V_1 \cup V_2 \cup \cdots \cup V_k, E)$ such that the degree of v_{jq} is d_{jq} for every q, $1 \leqq q \leqq p_j$.

(3) The time complexity of above (1) through (2) is $O(|V| + k^3)$, where $|V| = \Sigma^k_{j=1} p_j$.

In the following sections, $S = (s_1, s_2, \cdots, s_k)$ is called a k-partite multigraphical sequence set if S is a degree sequence set of a k-partite multigraph.

1. Introduction

The subject of this paper is the problem of finding an algorithm of constructing a k-partite multigraph from a k-partite multigraphical sequence set : " For a given integer constant k which satisfies k≧3, and, for k given non-negative integer sequences s_1, s_2, ····, s_k, s_j : d_{j1}, d_{j2}, ····, $d_{j,pj}$ (p_j≧1) for every j, 1≦j≦k, decide that whether S= (s_1, s_2, ····, s_k) is a k-partite multigraphical sequence set. If S is so then construct a k-partite multigraph G= (V_1 U V_2 U····U V_k, E) from it ", where V_j= {v_{j1}, v_{j2}, ····, $v_{j,pj}$} for every j, 1≦j≦k, and, for every q, 1≦q≦p_j, the degree of v_{jq} is d_{jq}. Set $x_j=\Sigma^{pj}_{q=1} d_{jq}$ for every j, 1≦j≦k.

In this paper, show that the k-partite multigraph construction problem (kMC-problem, for short) can be solved in linear time.

The problem of finding an algorithm of constructing a (multi) graph from a (multi) graphical sequence, is solved in [1][2][3][5]. In them, a polynomial time algorithm was given by Havel and Hakimi. The problem of finding an algorithm of constructing a bipartite (multi) graph from a bipartite (multi) graphical sequence set, is solved in [6]. In it, for a bipartite multigraph construction, a linear time algorithm was given, and, for a bipartite graph construction, a polynomial time algorithm was given.

In this paper, an O (|V|+k^3) algorithm of solving the kMC-problem is given, where |V|=$\Sigma^k_{j=1} p_j$. In the following sections, the following (1) through (2) will be discussed :

(1) Show a condition C such that a non-negative integer sequence set S= (s_1, s_2, ····, s_k) is a k-partite multigraphical sequence set if and only if C holds.

(2) Propose an O (|V|+k^3) algorithm satisfying the following (i) through (i i) :

(i) Decide that whether a given non-negative integer sequence set S= (s_1, s_2, ····, s_k) is a k-partite multigraphical sequence set.

(i i) If S= (s_1, s_2, ····, s_k) is a k-partite multigraphical sequence set then construct a k-partite multigraph G= (V_1 U V_2 U····U V_k, E) such that V_j= {v_{j1}, v_{j2}, ····, $v_{j,pj}$} for every j, 1≦j≦k, and such that, for every q, 1≦q≦p_j, the degree of v_{jq} is d_{jq}.

2. Preliminaries

A graph G= (V, E) consists of a finite set of vertices V and finite set of edges E such that each element of E is an unordered pair of distinct elements of V : E= {(u,v) | u,v∈V} .

For a given integer constant k which satisfies k≧3, and a graph G= (V, E) , G is called a k-partite (multi) graph if the following (1) through (3) are satisfied :

(1) V=V_1 U V_2 U····U V_k,

(2) For any two integer h and j, 1≦h≦k, 1≦j≦k, h≠j, $V_h \cap V_j = \Phi$ is satisfied, and

(3) For any edge e=(u,v), if u∈V_j then v∉V_j is satisfied, where 1≦j≦k.

For an edge e=(u,v), u (v, respectively) is <u>adjacent</u> to v (u) , u (v) is <u>incident</u> to e, and e is incident to v (u) . If u=v then the edge e is called a <u>self-loop</u>. For two edges e_1=(u,v) and e_2=(u',v'), e_1 and e_2 are called <u>multiple edges</u> if and only if e_1≠e_2, u=u' and v=v' hold. For a graph G, G is called a <u>multigraph</u> if G contains some multiple edges and no self-loop. For a graph G, G is called a <u>simple graph</u> (<u>graph</u>, for short) if G contains no multiple edge and no self-loop. For a vertex v∈V, a number of edges being incident to v, is called a <u>degree</u> of v and it is denoted by <u>deg(v)</u>.

A non-negative integer sequence set S= (s_1, s_2, ····, s_k) is a k-partite multigraphical sequence set if, for every j, 1≦j≦k, all vertices of V_j can be labeled v_{j1}, v_{j2}, ····, $v_{j,pj}$ such that the degree of v_{jq} is d_{jq} for every q, 1≦q≦p_j, where s_j : d_{j1}, d_{j2}, ····, $d_{j,pj}$, p_j≧1.

Set $x_j=\Sigma^{pj}_{q=1} d_{jq}$ for every j, 1≦j≦k.

3. Necessary and Sufficient Condition of a k-partite Multigraphical Sequence Set

In this section, discuss the condition C such that $S=(s_1, s_2, \cdots, s_k)$ is a k-partite multigraphical sequence set if and only if C holds, where s_j : $d_{j1}, d_{j2}, \cdots, d_{j,pj}$ ($p_j \geqq 1$ and $x_j \geqq 1$) for every j, $1 \leqq j \leqq k$, is a given non-negative integer sequence.

Such the condition C is obtained by the following theorem.

Theorem 1. For every j, $1 \leqq j \leqq k$, suppose that s_j : $d_{j1}, d_{j2}, \cdots, d_{j,pj}$, $p_j \geqq 1$, $x_j \geqq 1$, is a non-negative integer sequence, and suppose that $x_1 \leqq x_2 \leqq \cdots \leqq x_k$. Then $S=(s_1, s_2, \cdots, s_k)$ is a k-partite multigraphical sequence set if and only if the following condition (1) through (2) are satisfied : (1) $\Sigma^k{}_{j=1}x_j$ is an even number, and (2) $\Sigma^{k-1}{}_{j=1}x_j \geqq x_k$.

In the following, show the proof of Theorem 1.

Suppose that $S=(s_1, s_2, \cdots, s_k)$ is a k-partite multigraphical sequence set. Then it is clear that above (1) and (2) hold.

Inversely, suppose that there are k non-negative integer sequences $s_1, s_2, \cdots, s_k$, satisfying above (1) and (2). Then the following proposition can be obtained.

Proposition 1. For k non-negative integer sequences $s_1, s_2, \cdots, s_k$, such that $\Sigma^k{}_{j=1}x_j$ is an even number, assume that $\Sigma^{k-1}{}_{j=1}x_j \geqq x_k$ holds, where s_j : $d_{j1}, d_{j2}, \cdots, d_{j,pj}$ ($p_j \geqq 1$, $x_j \geqq 1$) for every j, $1 \leqq j \leqq k$. Set $x'_1 = x_1 - t_1$, $x'_2 = x_2 - t_2$, $\cdots$, $x'_{k-1} = x_{k-1} - t_{k-1}$ and $\Sigma^{k-1}{}_{j=1}t_j = x_k$, and let f : $b_1, b_2, \cdots, b_{k-1}$, be a sequence which is a result of sorting a sequence f' : $x'_1, x'_2, \cdots, x'_{k-1}$, and which satisfies $b_1 \leqq b_2 \leqq \cdots \leqq b_{k-1}$.

Then there is a sequence g : $t_1, t_2, \cdots, t_{k-1}$, which satisfies $\Sigma^{k-2}{}_{j=1}b_j \geqq b_{k-1}$ and which satisfies that $\Sigma^{k-1}{}_{j=1}b_j$ is an even number.

Proof. Consider two sequences f and g being made by using the following algorithm.

Algorithm A.

Begin

1. For every j, $1 \leqq j \leqq k$, $x'_j \leftarrow x_j$ and $t_j \leftarrow 0$; $x'_0 \leftarrow 0$;
2. For $j=k-2, 0, -1$ do begin
   (1) $y \leftarrow x'_{j+1} - x'_j$;
   If $\{x'_k < y \cdot (k-1-j)\}$ then begin $q \leftarrow j$; go to Step 3 end ;
   (2) For $r=k-1, j+1, -1$ do begin $t_r \leftarrow t_r + y$; $x'_r \leftarrow x'_r - y$ end ;
   $x'_k \leftarrow x'_k - y \cdot (k-1-j)$; If $\{x'_k = 0\}$ then halt end ;
3. $y \leftarrow \mathrm{div}\,(x'_k/(k-1-q))$; $x'_k \leftarrow \mathrm{mod}\,(x'_k/(k-1-q))$;
4. For $r=k-1, q+1, -1$ do begin $t_r \leftarrow t_r + y$; $x'_r \leftarrow x'_r - y$ end ; $r \leftarrow r-1$;
5. while $\{x'_k > 0\}$ do begin $t_r \leftarrow t_r + 1$; $x'_r \leftarrow x'_r - 1$; $x'_k \leftarrow x'_k - 1$; $r \leftarrow r-1$ end ;
6. Make a sequence f : $b_1, b_2, \cdots, b_{k-1}$, which is a result of sorting a sequence f' : $x'_1, x'_2, \cdots, x'_{k-1}$, and which satisfies $b_1 \leqq b_2 \leqq \cdots \leqq b_{k-1}$;
7. Make two sequences g_1 : $t'_1, t'_2, \cdots, t'_{k-1}$, and g_2 : $V'_1, V'_2, \cdots, V'_{k-1}$, satisfying the following : Assume that $b_r \leftarrow x'_h$ ($1 \leqq r \leqq k-1$, $1 \leqq h \leqq k-1$) holds by the sorting of step 6. Then, $t'_r \leftarrow t_h$ and $V'_r \leftarrow V_h$ are satisfied ;
8. For every j, $1 \leqq j \leqq k-1$, $x_j \leftarrow b_j$, $t_j \leftarrow t'_j$ and $V_j \leftarrow V'_j$

End. (Algorithm A terminates.)

Suppose that $x_{k-2}=0$. Then $x_k = x_{k-1}$ holds since $x_k \geqq x_{k-1}$ and $x_k \leqq \Sigma^{k-1}{}_{j=1}x_j = x_{k-1}$ are satisfied.

Thus $b_j = x'_j = 0$ for every j, $1 \leqq j \leqq k-1$, and, therefore, $\Sigma^{k-2}_{j=1} b_j \geqq b_{k-1}$ holds.

Suppose that $x_{k-2} > 0$. Assume that Algorithm A halts at Step 2 or that it executes Step 4 and and does not execute Step 5. Then $b_j = x'_j$ for every j, $1 \leqq j \leqq k-1$, and $b_{k-1} = b_{k-2}$ holds. Hence $b_{k-1} \leqq \Sigma^{k-2}_{j=1} b_j$ holds.

Assume that Algorithm A executes Step 5. If $b_{k-1} = b_{k-2}$ then it is clear that $b_{k-1} \leqq \Sigma^{k-2}_{j=1} b_j$ holds. Assume that $b_{k-1} > b_{k-2}$ holds. Then it is clear that $b_{k-1} = b_{k-2} + 1$ holds by the behavior of Algorithm A. If $b_{k-3} = 0$ then $\Sigma^{k}_{j=1} x_j = \Sigma^{k-1}_{j=1} b_j + 2x_k$ is an odd number, a contradiction. Thus $b_{k-3} \geqq 1$, and, therefore $b_{k-1} \leqq \Sigma^{k-2}_{j=1} b_j$ holds.

Hence, by above discussion, $b_{k-1} \leqq \Sigma^{k-2}_{j=1} b_j$ is satisfied for every b_j, $1 \leqq j \leqq k-1$, which is made by Algorithm A.

Thus Proposition 1 can be proved. Q. E. D.

By Proposition 1, the following proposition is obtained.

<u>Proposition 2.</u> For three non-negative integer sequences s_1, s_2, s_3, such that $\Sigma^{3}_{j=1} x_j$ is an even number, assume that $x_1 + x_2 \geqq x_3$ holds, where s_j : $d_{j1}, d_{j2}, \cdots, d_{j,p_j}$ ($p_j \geqq 1$, $x_j \geqq 1$) for every j, $1 \leqq j \leqq 3$. Set $x'_1 = x_1 - t_1$, $x'_2 = x_2 - t_2$ and $t_1 + t_2 = x_3$, and let f : b_1, b_2, be a sequence which is a result of sorting a sequence f' : x'_1, x'_2, and which satisfies $b_1 \leqq b_2$. Then there is a sequence g : t_1, t_2, which satisfies $b_1 = b_2$. Q. E. D.

By using Proposition 1 and 2, it is easy to prove that, for k non-negative integer sequences $s_1, s_2, \cdots, s_k$, $1 \leqq x_1 \leqq x_2 \leqq \cdots \leqq x_k$, $S = (s_1, s_2, \cdots, s_k)$ is a k-partite multigraphical sequence set if the following conditions (1) through (2) are satisfied :

(1) $\Sigma^{k}_{j=1} x_j$ is even number, and (2) $\Sigma^{k-1}_{j=1} x_j \geqq x_k$.

The proof is obtained by the following algorithm easily.

<u>Algorithm B.</u>

<u>Begin</u>

1. k ← a given integer constant n which satisfies $n \geqq 3$;
2. <u>while</u> $\{k \geqq 2\}$ <u>do</u> <u>begin</u> perform Algorithm A ; k ← k−1 <u>end</u>

<u>End.</u> (Algorithm C terminates.)

It is clear that, by performing Step 2 at most k−1 times, a k-partite multigraph G which is constructed from a k-partite multigraphical sequence set $S = (s_1, s_2, \cdots, s_k)$, can be constructed.

By above discussion, Theorem 1 has been proved.

4. <u>Data Structure and Algorithm</u>

By Theorem 1, an algorithm of solving the BMC-problem, can be obtained directly. In this section, discuss such an algorithm.

4.1 <u>Data Structure</u>

Suppose that $V_j = \{v_{j1}, v_{j2}, \cdots, v_{j,p_j}\}$, $p_j \geqq 1$, for every j, $1 \leqq j \leqq k$. Use an array ADJLIST containing k listheads. Each listhead represents a set of vertices V_j. For every j,

$0 \leqq j \leqq k$, j-th listhead has the form [NUM, PNT], where NUM contains a current x_j and PNT is a pointer field. The nodes in the linked lists have the form [VTX, DEG, LINK], where VTX is a vertex number, DEG is a current degree of a vertex, LINK is a pointer field and $x_0 = 0$. Use an array ELIST containing a listhead. The listhead represents a set of edges of the final k-partite multigraph which is constructed from $S = (s_1, s_2, \cdots, s_k)$. The nodes in the linked list have the form [u, v, ENM, ELK], where u is a vertex of V_j ($1 \leqq j \leqq k$), v is a vertex of V_h ($1 \leqq h \leqq k$, $h \neq j$), ENM is a number of edge (u,v), satisfying $ENM \geqq 1$, and ELK is a pointer field.

For example, suppose that $k=3$, that s_1 : 5, 5, 5, that s_2 : 6, 6, 6, that s_3 : 6, 4, 4, 5, and that a final k-partite multigraph is shown in Fig.1. Then the data structure is the following.

ADJLIST

V_0 [0, Λ], V_1 [15, →] → [v_{11}, 5, →] → [v_{12}, 5, →] → [v_{13}, 5, Λ]

V_2 [18, →] → [v_{21}, 6, →] → [v_{22}, 6, →] → [v_{23}, 6, Λ]

V_3 [19, →] → [v_{31}, 6, →] → [v_{32}, 4, →] → [v_{33}, 4, →] → [v_{34}, 5, Λ]

ELIST [→] → [v_{31}, v_{11}, 2, →] → [v_{31}, v_{12}, 2, →] → [v_{31}, v_{13}, 2, →] → [v_{32}, v_{12}, 1, →]
→ [v_{32}, v_{13}, 1, →] → [v_{32}, v_{22}, 1, →] → [v_{32}, v_{23}, 1, →] → [v_{33}, v_{21}, 2, →]
→ [v_{33}, v_{22}, 1, →] → [v_{33}, v_{23}, 1, →] → [v_{34}, v_{21}, 1, →] → [v_{34}, v_{22}, 2, →]
→ [v_{34}, v_{23}, 2, →] → [v_{21}, v_{11}, 2, →] → [v_{21}, v_{12}, 1, →] → [v_{22}, v_{11}, 1, →]
→ [v_{22}, v_{13}, 1, →] → [v_{23}, v_{12}, 1, →] → [v_{23}, v_{13}, 1, Λ]

In the following of this paper, for every j, $1 \leqq j \leqq k$, NUM of a listhead V_j is denoted by NUM(V_j), PNT of a listhead V_j is denoted by POINT(V_j), VTX of a vertex v_{jh} ($1 \leqq h \leqq p_j$) is denoted by VTX(v_{jh}), DEG of a vertex v_{jh} is denoted by DEG(v_{jh}) and LINK of a vertex v_{jh} is denoted by LINK(v_{jh}), a listhead of ELIST is denoted by PTR, ENM of an edge e=(u,v) is denoted by ENM(e) and ELK of an edge e=(u,v) is denoted by ELK(e).

4.2 Algorithm

Suppose that $x_1 \leqq x_2 \leqq \cdots \leqq x_k$.

Algorithm kMGC.

Begin

1. perform Procedure Prep ;
2. If (status≠0) then go to Step 5 ;
3. while ($k \geqq 2$) do begin
   perform Procedure Edgdec ; perform Procedure Edgadd ; k ← k−1 end ;
4. A k-partite multigraph $G = (V_1 \cup V_2 \cup \cdots \cup V_k, E)$ with, for every j, $1 \leqq j \leqq k$, $\deg(v_{jq}) = d_{jq}$ for every q, $1 \leqq q \leqq p_j$, is constructed ; halt ;
5. A sequence set $S = (s_1, s_2, \cdots, s_k)$ is not a k-partite multigraphical sequence set

End. (Algorithm kMGC terminates.)

Procedure Prep.

Begin

1. NUM(V_j) ← $\Sigma^{p_j}_{q=1} d_{jq}$ and POINT(V_j) ← Λ for every j, $1 \leqq j \leqq k$; PTR ← Λ ; status ← 0 ;
2. For j=1, k do begin
   For q=1, p_j do begin
   If ($d_{jq} > 0$) then begin
   LINK(v_{jq}) ← POINT(V_j) ; DEG(v_{jq}) ← d_{jq} ;

$VTX(v_{jq}) \leftarrow v_{jq}$; $POINT(V_j) \leftarrow VTX(v_{jq})$ end end end ;

3. $x \leftarrow \Sigma^{k}_{j=1} NUM(V_j)$; $y \leftarrow x - NUM(V_k)$; If $\{NUM(V_k) > y\}$ then status $\leftarrow 1$

End. (Procedure Prep terminates.)

Procedure Edgdec.

Begin

1. For every j, $1 \leqq j \leqq k-1$, $t_j \leftarrow 0$; $NUM(V_0) \leftarrow 0$; $POINT(V_0) \leftarrow \Lambda$;
2. For $j = k-2, 0, -1$ do begin
   (1) $y \leftarrow NUM(V_{j+1}) - NUM(V_j)$;
   If $\{NUM(V_k) < y \cdot (k-1-j)\}$ then begin $q \leftarrow j$; go to Step 3 end ;
   (2) For $r = k-1, j+1, -1$ do begin $t_r \leftarrow t_r + y$; $NUM(V_r) \leftarrow NUM(V_r) - y$ end ;
   $NUM(V_k) \leftarrow NUM(V_k) - y \cdot (k-1-j)$; If $\{NUM(V_k) = 0\}$ then halt end ;
3. $y \leftarrow \mathrm{div}\,(NUM(V_k)/(k-1-q))$; $NUM(V_k) \leftarrow \mathrm{mod}\,(NUM(V_k)/(k-1-q))$;
4. For $r = k-1, q+1, -1$ do begin $t_r \leftarrow t_r + y$; $NUM(V_r) \leftarrow NUM(V_r) - y$ end ; $r \leftarrow r-1$;
5. while $\{x'_k > 0\}$ do begin
   $t_r \leftarrow t_r + 1$; $NUM(V_r) \leftarrow NUM(V_r) - 1$; $NUM(V_k) \leftarrow NUM(V_k) - 1$; $r \leftarrow r-1$ end ;
6. Make a sequence f' : $b_{11}, b_{12}, \cdots, b_{1,k-1}$, which is a result of sorting a sequence f : $NUM(V_1), NUM(V_2), \cdots, NUM(V_{k-1})$, and which satisfies $b_{11} \leqq b_{12} \leqq \cdots \leqq b_{1,k-1}$;
7. Make two sequences g_1 : $t'_1, t'_2, \cdots, t'_{k-1}$, and g_2 : $b_{21}, b_{22}, \cdots, b_{2,k-1}$, satisfying the following : Assume that $b_{1r} \leftarrow NUM(V_h)$ $(1 \leqq r \leqq k-1,\ 1 \leqq h \leqq k-1)$ holds by the sorting of step 6. Then, $t'_r \leftarrow t_h$ and $b_{2r} \leftarrow POINT(V_h)$ are satisfied ;
8. For every j, $1 \leqq j \leqq k-1$, $NUM(V_j) \leftarrow b_{1j}$, $t_j \leftarrow t'_j$ and $POINT(V_j) \leftarrow b_{2j}$

End. (Procedure Edgdec terminates.)

Procedure Edgadd.

Begin

For $j = k-1, 1, -1$ do begin
   $u \leftarrow POINT(V_k)$; $v \leftarrow POINT(V_j)$;
   while $\{t_j > 0\}$ do begin
      $y \leftarrow \min\,(t_j, DEG(u), DEG(v))$; $ENM(e) \leftarrow y$, where $e = (u,v)$; $ELK(e) \leftarrow PTR$;
      $PTR \leftarrow u$; $DEG(u) \leftarrow DEG(u) - y$; $DEG(v) \leftarrow DEG(v) - y$; $t_j \leftarrow t_j - y$;
      If $\{d'_{kh} = 0\}$ then begin $POINT(V_k) \leftarrow LINK(u)$; $u \leftarrow POINT(V_k)$ end ;
      If $\{d'_{jr} = 0\}$ then begin $POINT(V_j) \leftarrow LINK(v)$; $v \leftarrow POINT(V_j)$ end end end

End. (Procedure Edgadd terminates.)

5. Example

Set $d_{11}=5$, $d_{12}=5$, $d_{13}=5$, $d_{21}=6$, $d_{22}=6$, $d_{23}=6$, $d_{31}=6$, $d_{32}=4$, $d_{33}=4$ and $d_{34}=5$ ($p_1=3$, $p_2=3$ and $p_3=4$) . Then it is clear that $S=(s_1, s_2, s_3)$ is a k-partite multigraphical sequence set.

1. By Step 1, the following data structure is obtained.

V_1 [15, →] → [v_{13}, 5, →] → [v_{12}, 5, →] → [v_{11}, 5, Λ]

V_2 [18, →] → [v_{23}, 6, →] → [v_{22}, 6, →] → [v_{21}, 6, Λ]

V_3 [19, →] → [v_{34}, 5, →] → [v_{33}, 4, →] → [v_{32}, 4, →] → [v_{31}, 6, Λ]

2. By Step 3, the following (1) through (3) are performed :
   (1) By performing Procedure Edgdec, $t_2=11$, $t_1=8$, $NUM(V_2)=7$ and $NUM(V_1)=7$ are obtained.
   (2) By performing Procedure Edgadd, the following data structure is constructed.

ELIST [→] → [v_{31}, v_{12}, 3, →] → [v_{31}, v_{13}, 3, →] → [v_{32}, v_{13}, 2, →] → [v_{32}, v_{22}, 2, →] → [v_{33}, v_{22}, 3, →] → [v_{33}, v_{23}, 1, →] → [v_{34}, v_{23}, 5, Λ]

(3) Then $k=2$ and the following data structure are obtained.

V_1 [7, →] → [v_{12}, 2, →] → [v_{11}, 5, Λ] , V_2 [7, →] → [v_{22}, 1, →] → [v_{21}, 6, Λ]

3. By Step 3, the following (1) through (3) are performed :

(1) By performing Procedure Edgdec, $t_1=7$ and NUM(V_1)=0 are obtained.

(2) By performing Procedure Edgadd, the following data structure is constructed.

ELIST [→] → [v_{21}, v_{11}, 5, →] → [v_{21}, v_{12}, 1, →] → [v_{22}, v_{12}, 1, →] → [v_{31}, v_{12}, 3, →]
→ [v_{31}, v_{13}, 3, →] → [v_{32}, v_{13}, 2, →] → [v_{32}, v_{22}, 2, →] → [v_{33}, v_{22}, 3, →]
→ [v_{33}, v_{23}, 1, →] → [v_{34}, v_{23}, 5, Λ]

(3) Then $k=1$ and the following data structure are obtained : V_1 [0, Λ] .

4. Algorithm kMGC terminates, and a k-partite multigraph being shown in Fig.2, is obtained.

The final graph G satisfies $deg(v_{11})=5$, $deg(v_{12})=5$, $deg(v_{13})=5$, $deg(v_{21})=6$, $deg(v_{22})=6$, $deg(v_{23})=6$, $deg(v_{31})=6$, $deg(v_{32})=4$, $deg(v_{33})=4$ and $deg(v_{34})=5$.

6. Time complexity

In this section, discuss the time complexity of Algorithm kMGC.

1. The time complexity of Step 1 is $O(|V|)$, where $|V|=\Sigma^k_{j=1}p_j$.
2. For Step 3, discuss the following [1] through [2].

[1] For Procedure Edgdec, discuss the following (1) through (4) :

(1) The time complexity of Step 1 is $O(\Sigma^k_{j=2}j)=O(k^2)$.

(2) The time complexity of Step 2 is $O(\Sigma^k_{j=2}j^2)=O(k^3)$.

(3) The time complexity of Step 4 and 5 is $O(k)$.

(4) The time complexity of Step 6 through 8 is $O(\Sigma^k_{j=2}j\cdot\log_2 j)$.

Thus the time complexity of Procedure Edgdec is $O(k^3)$.

[2] For Procedure Edgadd, discuss the following discussion.

$A_{k,k-1}$,	$A_{k,k-2}$,	····,	A_{k2} ,	A_{k1}	$V'_{k,k-1}$,	$V'_{k,k-2}$,	····,	V'_{k2} ,	V'_{k1}
$A_{k-1,k}$,	$A_{k-1,k-2}$,	····,	$A_{k-1,2}$,	$A_{k-1,1}$	$V'_{k-1,k}$,	$V'_{k-1,k-2}$,	····,	$V'_{k-1,2}$,	$V'_{k-1,1}$
.	.		.	.	.	.		.	.
.	.		.	.	.	.		.	.
A_{2k} ,	$A_{2,k-1}$,	····,	A_{23} ,	A_{21}	V'_{2k} ,	$V'_{2,k-1}$,	····,	V'_{23} ,	V'_{21}
A_{1k} ,	$A_{1,k-1}$,	····,	A_{12} ,	A_{11}	V'_{1k} ,	$V'_{1,k-1}$,	····,	V'_{12} ,	V'_{11}

Table.1 Table.2

For above two tables, let A_{jq} ($1\leqq j\leqq k$, $1\leqq q\leqq k$, $j\neq q$) be a set of edges of a final graph satisfying $A_{jq}=\{(u,v)\mid u\in V_j \text{ and } v\in V_q\}$. Then $A_{jq}=A_{qj}$ holds, and let V'_{jq} be a set of vertices of V_j satisfying $u\in V_j$ and $v\in V_q$ for an edge $e=(u,v)\in A_{jq}$. Two sets that exist side by side in Table.2 share at most one vertex.

Hence $\Sigma^k_{j=1}\Sigma^k_{q=1}|V'_{jq}|\leqq|V|+k\cdot(k-2)$ holds and Edge addition is performed at $\Sigma^k_{j=1}\Sigma^{j-1}_{q=1}(|V'_{jq}|+|V'_{qj}|-1)\leqq|V|+k\cdot(k-2)-k\cdot(k-1)/2=O(|V|+k^2)$ times.

Thus the time complexity of Procedure Edgadd is $O(|V|+k^2)$.

By above [1] and [2], the time complexity of Step 3 is $O(|V|+k^3)$.

By above discussion 1 and 2, the time complexity of Algorithm kMGC is $O(|V|+k^3)$.

7. Conclusion

In this paper, a k-partite multigraph construction algorithm which performs the following (1) through (2), is obtained, where $k \geqq 3$:

(1) For a given sequence set $S = (s_1, s_2, \cdots, s_k)$, $s_j : d_{j1}, d_{j2}, \cdots, d_{j,p_j}$, for every j, $1 \leqq j \leqq k$, decide that whether S is a k-partite multigraphical sequence set, and

(2) If S is so then construct a k-partite multigraph from S. Then the result is $O(|V| + k^3)$, where $|V| = \Sigma^{k}_{j=1} p_j$.

I want to find an improved algorithm of Procedure Edgdec, and an algorithm of constructing an n-partite graph from an n-partite graphical sequence set $S = (s_1, s_2, \cdots, s_n)$ for further investigation, where n is a given integer which satisfies $n \geqq 3$.

References

[1] P.Erdos and T.Gallai, Graphs with prescribed degrees of vertices (Hungarian), Mat, Lapok 11(1960), 264-274

[2] 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), 496-506

[3] V.Havel, A remark on the existence of finite graphs (Czech), Casopis Pest, Mat, 80(1955), 477-480

[4] H.Frank and W.Chou, Connectivity considerations in the design of survivable networks, IEEE Trans. Circuit Theory, CT-17.(1970), 486-490

[5] M.Behzad, G.Chartrand and L.Lesnik-Foster, "Graphs and Digraphs," Prindle, Weber and Schmidt, (1979)

[6] M.Takahashi, An Algorithm of Constructing a Bipartite Graph from a Bipartite Graphical Sequence Set, Information Processing Society of Japan, Tech. Rep. 92-AL27-5.

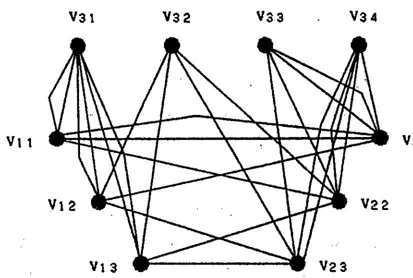

Fig.1

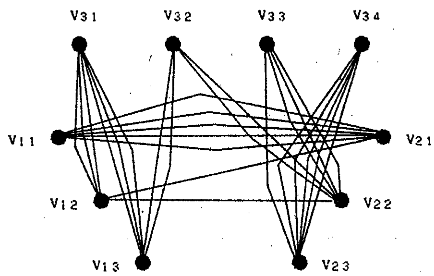

Fig.2