アルゴリズム 30－4
(1992. 11. 20)

k組グラフ的次数列集合からk組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

〒443 愛知県蒲郡市西迫町馬乗50-2

あらまし　本稿では、$k \geqq 2$ なる任意の整定数 k について、k 個の次数列 s_j : d_{j1}, d_{j2}, ‥‥, d_{jp_j} ($1 \leqq j \leqq k$) が与えられた時、次の（１）（２）について述べる。

（１）　S＝(s_1, s_2, ‥‥, s_k)　がk組グラフ的次数列集合であるための必要十分条件を示す。

（２）　その必要十分条件より、以下の ①～③ を充たすようなアルゴリズムを提案する。

① S から、k組グラフ G＝(V, E) を構成できるかどうかを判定する。

② もし、そのようなk組グラフが構成できるなら、S はk組グラフ的次数列集合である。

③ 上記 ①② の時間複雑度が O ($k|V|^2$) である。ただし、$|V| = \Sigma^k_{j=1} p_j$ である。

和文キーワード　k組グラフ，k組グラフ的次数列集合，必要十分条件，多項式時間アルゴリズム，データ構造

An Algorithm of Constructing a k-partite Graph from a k-partite Graphical Sequence Set

Masaya Takahashi

Department of Electronic Engineering, Aichi College of Technology

Nishihasama-cho, Gamagori-shi, Aichi-ken, 443 Japan

Abstract　In this paper, when k ($\geqq 2$) degree sequences s_j : d_{j1}, d_{j2}, ‥‥, d_{jp_j}, for every j, $1 \leqq j \leqq k$, is given, discuss the following (1) and (2) :

(1)　Find the necessary and sufficient condition C of a k-partite graphical sequence set S = (s_1, s_2, ‥‥, s_k) .

(2)　By the condition C, propose the algorithm satisfying the following (i) through (iii) :

(i)　Decide that whether a k-partite graph can be constructed from S.

(ii)　If such the graph can be constructed then S is a k-partite graphical sequence set.

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

英文 key words　k-partite graph, k-partite graphical sequence set, necessary and sufficient condition, polynomial time algorithm, data structure

1. Introduction

The subject of this paper is the problem of finding an algorithm of constructing a k-partite graph from a k-partite graphical sequence set : " For a given integer constant k which satisfies $k \geqq 2$, and, for k given non-negative integer sequences $s_1, s_2, \cdots, s_k$, s_j : $d_{j1}, d_{j2}, \cdots, d_{j,pj}$ ($p_j \geqq 1$) for every j, $1 \leqq j \leqq k$, if a k-partite graph $G = (V_1 \cup V_2 \cup \cdots \cup V_k, E)$ such that, for every q, $1 \leqq q \leqq p_j$, the degree of v_{jq} is d_{jq} for every j, $1 \leqq j \leqq k$, is constructed from the sequences, then $S = (s_1, s_2, \cdots, s_k)$ is a k-partite graphical sequence set ", where $V_j = \{v_{j1}, v_{j2}, \cdots, v_{j,pj}\}$ for every j, $1 \leqq j \leqq k$. Set $x_j = \Sigma^{pj}_{q=1} d_{jq}$ for every j, $1 \leqq j \leqq k$.

In this paper, show that the k-partite graph construction problem (kC-problem, for short) can be solved in polynomial 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 is given by Havel and Hakimi. The problem of finding an algorithm of constructing a bipartite multigraph from a bipartite multigraphical sequence set, is solved in [6]. In it, a linear time algorithm is given. The problem of finding an algorithm of constructing a k-partite multigraph from a k-partite multigraphical sequence set, is solved in [7]. In it, a linear time algorithm is given.

In this paper, for a given integer constant k which satisfies $k \geqq 2$, an $O(k|V|^2)$ algorithm of solving the kC-problem, is given, where $|V| = \Sigma^k_{j=1} p_j$.

In the following sections, the following (1) and (2) will be discussed :

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

(2) By the condition C, propose an algorithm satisfying the following (i) through (iii) :

(i) Decide that whether a k-partite graph $G = (V_1 \cup V_2 \cup \cdots \cup V_k, E)$ such that $V_j = \{v_{j1}, v_{j2}, \cdots, v_{j,pj}\}$ for every j, $1 \leqq j \leqq k$, and such that, for every q, $1 \leqq q \leqq p_j$, the degree of v_{jq} is d_{jq}, can be constructed from $S = (s_1, s_2, \cdots, s_k)$,

(ii) If such a k-partite graph can be constructed then $S = (s_1, s_2, \cdots, s_k)$ is a k-partite graphical sequence set, and

(iii) The time complexity of above (i) and (ii) is $O(k|V|^2)$.

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 : $E = \{(u,v) \mid u,v \in V\}$.

For a given integer constant k which satisfies $k \geqq 2$, and a graph $G = (V, E)$, G is called a k-partite graph if the following (1) through (3) are satisfied :

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

(2) For any two integers 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 $1 \leqq j \leqq k$.

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

A non-negative integer sequence set $S=(s_1, s_2, \cdots, s_k)$ is a k-partite graphical sequence set if, for every j, $1 \leqq j \leqq k$, all vertices of V_j can be labeled v_{j1}, v_{j2}, $\cdots$, $v_{j,pj}$, such that the degree of v_{jq} is d_{jq} for every q, $1 \leqq q \leqq p_j$, where s_j : $d_{j1}, d_{j2}, \cdots, d_{j,pj}$, $p_j \geqq 1$. Set $x_j = \Sigma^{pj}_{q=1} d_{jq}$ for every j, $1 \leqq j \leqq k$.

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

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

Let f : $b_1, b_2, \cdots, b_p$, be a sequence which is a result of sorting a sequence s : $d_{11}, \cdots, d_{1,p1}, d_{21}, \cdots, d_{2,p2}, \cdots, d_{k1}, \cdots, d_{k,pk}$, and which satisfies $b_1 \leqq b_2 \leqq \cdots \leqq b_p$, where $p=\Sigma^k_{j=1} p_j$.

Make two sequences g_1 : $u_1, u_2, \cdots, u_p$, and g_2 : $n_1, n_2, \cdots, n_p$, satisfying the following : Assume that $b_r \leftarrow d_{jh}$ ($1 \leqq r \leqq p$, $1 \leqq j \leqq k$, $1 \leqq h \leqq p_j$) holds by above sorting. Then, $u_r \leftarrow v_{jh}$ and $n_r \leftarrow j$ are satisfied. Set $V=\{u_1, u_2, \cdots, u_p\}$ (i.e., set $V = V_1 \cup V_2 \cup \cdots \cup V_k$). Such the condition C is obtained by the following theorem.

Theorem 1. For every non-negative integer sequence s_j : $d_{j1}, d_{j2}, \cdots, d_{j,pj}$, $1 \leqq j \leqq k$, suppose that $d_{j1} \leqq d_{j2} \leqq \cdots \leqq d_{j,pj}$, that $p_j \geqq 1$ and that $1 \leqq d_{j,pj} \leqq p-p_j$, and suppose that $u_p \in V_r$ for some r, $1 \leqq r \leqq k$, and that $x_1 \leqq x_2 \leqq \cdots \leqq x_k$, where $p=\Sigma^k_{j=1} p_j$. Then f : $b_1, b_2, \cdots, b_p$, is a k-partite graphical sequence (i.e., $S=(s_1, s_2, \cdots, s_k)$ is a k-partite graphical sequence set) if and only if f' : $b'_1, b'_2, \cdots, b'_{p-1}$, is a k'-partite graphical sequence, where k' is an integer satisfying, $k'=k-1$ if $p_r = 1$, $k'=k$ otherwise, and f' is a sequence which is made by the following algorithm.

Algorithm A.

Begin

1. $x \leftarrow \Sigma^k_{j=1} x_j$; $b'_p \leftarrow b_p$; $r \leftarrow n_p$;
2. For $j=p-1, 1, -1$ do begin
   $b'_j \leftarrow b_j$; $h \leftarrow n_j$;
   If $(b'_p > 0)$ and $(h \neq r)$ then begin
     $x' \leftarrow x-2$; $x'_r \leftarrow x_r - 1$; $x'_h \leftarrow x_h - 1$;
     $z \leftarrow \min(r, h)$; $q \leftarrow \max(r, h)$;
     $y \leftarrow \max(x'_r, x'_h, x_1, \cdots, x_{z-1}, x_{z+1}, \cdots, x_{q-1}, x_{q+1}, \cdots, x_k)$;
     If $(y \leqq x'-y)$ then begin

$b'_j \leftarrow b'_j - 1$; $b'_p \leftarrow b'_p - 1$; $x \leftarrow x'$; $x_r \leftarrow x'_r$;
$x_h \leftarrow x'_h$ <u>end</u> <u>end</u> <u>end</u>

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

In the following, show the proof of Theorem 1.

Suppose that f' : $b'_1, b'_2, \cdots, b'_{p-1}$, is a k'-partite graphical sequence. There is a k'-partite graph $G_1 = (V_1, E_1)$ such that $\deg(u'_j) = b'_j$ holds for every j, $1 \leqq j \leqq p-1$.

Let $G = (V, E)$ be a new k-partite graph having $V = V_1 \cup \{u_p\}$ and $E = E_1 \cup A$, where $A = \{(u_p, u'_j) \mid$ every j satisfies $b'_j \neq b_j\}$.

For every vertex $u'_j \in G_1$, $1 \leqq j \leqq p-1$, assume that the label of u'_j is replaced to $u_j \in G$. Then, for G, $\deg(u_j) = b_j$ is satisfied for every j, $1 \leqq j \leqq p$.

Hence f : $b_1, b_2, \cdots, b_p$, is a k-partite graphical sequence.

Inversely, suppose that f : $b_1, b_2, \cdots, b_p$, is a k-partite graphical sequence. (i.e., suppose that there is a k-partite graph G such that $\deg(u_j) = b_j$ holds for every j, $1 \leqq j \leqq p$.)

Assume that G contains a vertex $u \in V_q$ for some q, $1 \leqq q \leqq k$, such that the following (1) and (2) are satisfied :

(1) $\deg(u) = b_p$, and

(2) For every j satisfying $b'_j \neq b_j$, there is an edge $e = (u, u_j)$.

Then a k"-partite graph $G - u$ has a sequence f' : $b'_1, b'_2, \cdots, b'_{p-1}$, and, therefore, f' is a k"-partite graphical sequence, where k" is an integer satisfying, $k'' = k - 1$ if $p_q = 1$, $k'' = k$ otherwise.

In the following, suppose that G does not contain a vertex $u \in V$ such that above (1) and (2) are satisfied. Then, for the vertex u_p, there are two sets of vertices U_1 and U_2 satisfying the following conditions (3) through (5) :

(3) For any vertex $u_j \in U_1$, $b'_j \neq b_j$ $(1 \leqq j \leqq p-1)$ is satisfied, and G does not have an edge (u_p, u_j),

(4) For any vertex $u_j \in U_2$, $b'_j = b_j$ is satisfied, and G has an edge (u_p, u_j),

and (5) $|U_1| = |U_2|$ is satisfied.

Suppose that $w_1 \in V_h$ $(1 \leqq h \leqq k,\ h \neq r)$ for any vertex $w_1 \in U_1$. For a vertex $w_2 \in U_2$, set $e_1 = (u_p, w_1)$ and $e_2 = (u_p, w_2)$. (It is clear that e_1 is not contained in G.) Since $\deg(w_1) \geqq \deg(w_2)$, there is a vertex $v' \notin V_h$ such that there is an edge $e_3 = (w_1, v')$ and such that there is not an edge $e_4 = (w_2, v')$. Then the following [1] and [2] are discussed.

[1] Assume that there is a vertex of $U_2 \cap V_h$. Let w_2 be any vertex of $U_2 \cap V_h$. Since $v' \notin V_h$, set $G' = G + \{e_1, e_4\} - \{e_2, e_3\}$. Then G' is a k-partite graph and has a same sequence f : $b_1, b_2, \cdots, b_p$, of G.

[2] Assume that there is not a vertex of $U_2 \cap V_h$. Let w_2 be any vertex of $U_2 \cap V_n$ $(1 \leqq n \leqq k,\ n \neq r,\ n \neq h)$. Then the following (1) and (2) are discussed.

(1) Assume that there is a vertex $v' \notin V_n$. Set $G' = G + \{e_1, e_4\} - \{e_2, e_3\}$. Then G' is a k-partite graph and has a same sequence f : $b_1, b_2, \cdots, b_p$, of G.

(2) Assume that $v' \in V_n$ is satisfied for every vertex v', and that there is a vertex $w_3 \in U_2 \cap V_q$ $(1 \leqq q \leqq k,\ q \neq r,\ q \neq h,\ q \neq n)$. Set $e_5 = (u_p, w_3)$, $e_6 = (w_3, v')$ and $G' = G + \{e_1, e_6\} - \{e_3, e_5\}$. Then G' is a k-partite graph and has a same sequence f :

b_1, b_2, ⋯, b_p, of G.

(3) Assume that $v' \in V_n$ is satisfied for every vertex v', and that there is not a vertex of $U_2 \cap V_q$ for every q, $1 \leqq q \leqq k$, $q \neq h$, $q \neq n$. Then the following proposition is obtained.

Proposition 1. For every j, $1 \leqq j \leqq k$, let $B_j = \{(u_p, v'') \mid v'' \in V_j\}$ be a set of edges in G, and $D_j = \{v'' \mid (u_p, v'') \in G$ and $v'' \in V_j\}$ be a set of vertices in G. Set $t_j = |B_j| = |D_j|$, $z = x_n - t_n$ and $a = \Sigma^k_{j=1}(x_j - t_j) - z$. (It is clear that $b_p = \Sigma^k_{j=1} t_j$ holds.) Then $z < a$ is satisfied.

Proof. Since G is a k-partite graph, $z \leqq a$ is satisfied. Assume that $z = a$ is satisfied. Let q be any integer satisfying $1 \leqq q \leqq k$, $q \neq h$ and $q \neq n$. Since $|U_2 \cap V_q| = \Phi$ holds, $b'_j \neq b_j$ holds for every vertex $u_j \in D_q$, $1 \leqq j \leqq p$. Similarly, $b'_j \neq b_j$ holds for every vertex $u_j \in D_h$, $1 \leqq j \leqq p$.

Set $D'_m = \{u_j \mid$ every j satisfies $b'_j \neq b_j\}$ and $t'_m = |D'_m|$ for every m, $1 \leqq m \leqq k$. Set $z' = x_n - t'_n$ and $a' = \Sigma^k_{j=1}(x_j - t'_j) - z'$. Then $t'_h \geqq t_h$ holds and $t'_q \geqq t_q$ holds for every q. If $t'_h > t_h$ holds or, for some q, $t'_q > t_q$ holds then $t'_n < t_n$ holds, and, therefore, $z' > a'$ is satisfied, a contradiction. (Contradict the behavior of Algorithm A.) Thus $t'_h = t_h$ and $D'_h = D_h$ hold, and $t'_q = t_q$ and $D'_q = D_q$ hold for every q.

Thus any edge $e \in B_q$ can not removed and t_h can not be increased since $z = a$.

Hence, since $b'_j \neq b_j$ holds for every vertex $u_j \in D_h = D'_h$, $1 \leqq j \leqq p$, w_1 does not become a vertex of $U_1 \cap V_h$ by the behavior of Algorithm A, a contradiction.

Hence $z < a$ is satisfied. Q. E. D.

By Proposition 1, there is an edge $e' = (w'_1, w'_2)$ such that $w'_1 \notin V_n$, $w'_2 \notin V_n$, $w'_1 \neq u_p$ and $w'_2 \neq u_p$ hold. Set $e_4 = (w_2, w'_1)$, $e_5 = (v', w'_2)$ and $G' = G + \{e_1, e_4, e_5\} - \{e_2, e_3, e'\}$. Then G' is a k-partite graph and has a same sequence f : b_1, b_2, ⋯, b_p, of G.

By repeating above operation [1] and [2] until $|U_1| = |U_2| = 0$, a k-partite graph containing the vertex $u_p \in V$ such that the following condition is satisfied, can be obtained : For every j satisfying $b'_j \neq b_j$, there is an edge $e = (u_p, u_j)$.

Then a k'-partite graph $G - u_p$ has a sequence set f' : b'_1, b'_2, ⋯, b'_{p-1}, and, therefore, f' is a k'-partite graphical sequence.

By above discussion, Theorem 1 has been proved.

4. Data Structure and Algorithm

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

4.1 Data Structure

Use two linked lists H_1 and H_2. Their data structures are the following (1) and (2) :

(1) H_2 represents the vertex u_p, and H_1 represents a set of vertices $V - \{u_p\}$.

(2) The nodes in the linked lists have the form [VTX, DEG, BLG, LINK], where VTX is a vertex number, DEG is a current degree of the vertex, BLG is a set of vertices V_j containing the vertex ($1 \leqq j \leqq k$) and LINK is a pointer field.

For every n, $1 \leqq n \leqq 2$, use an array LST_n containing p−1 listheads. Their data structures are the following (1) and (2):

(1) There are p−1 listheads. Each listhead represents a degree of vertices of $V - \{u_p\}$. For every r, $1 \leqq r \leqq p-1$, r-th element of the array indicates a node which represents a vertex v with deg(v) = r.

(2) The nodes in the linked lists are the same of the form of H_1 and H_2.

For example, suppose that s_1 : 1, 3, that s_2 : 2, 2, 2, that s_3 : 1, 2, 2, 3, that $V_1 = \{v_{11}, v_{12}\}$, that $V_2 = \{v_{21}, v_{22}, v_{23}\}$ and that $V_3 = \{v_{31}, v_{32}, v_{33}, v_{34}\}$. Then three sequences f : 1, 1, 2, 2, 2, 2, 2, 3, 3, g_1 : v_{31}, v_{11}, v_{32}, v_{33}, v_{21}, v_{22}, v_{23}, v_{34}, v_{12}, and g_2 : 3, 1, 3, 3, 2, 2, 2, 3, 1, are obtained. The data structures are the following.

H_1 [→] → [v_{34}, 3, 3, →] → [v_{23}, 2, 2, →] → [v_{22}, 2, 2, →] → [v_{21}, 2, 2, →] →
→ [v_{33}, 2, 3, →] → [v_{32}, 2, 3, →] → [v_{11}, 1, 1, →] → [v_{31}, 1, 3, Λ]

H_2 [→] → [v_{12}, 3, 1, Λ]

LST_1 : 1 [→] → [v_{11}, 1, 1, →] → [v_{31}, 1, 3, Λ], 3 [→] → [v_{34}, 3, 3, Λ],
2 [→] → [v_{23}, 2, 2, →] → [v_{22}, 2, 2, →] → [v_{21}, 2, 2, →] →
→ [v_{33}, 2, 3, →] → [v_{32}, 2, 3, Λ],
4 [Λ], 5 [Λ], 6 [Λ], 7 [Λ], 8 [Λ], where p = 9.

In the following of this paper, for every j, $1 \leqq j \leqq 2$, a pointer of a listhead of H_j is denoted by <u>POINT(H_j)</u>, and an r-th listhead of LST_n is denoted by <u>POINT$_n$(r)</u> ($1 \leqq n \leqq 2$). For every j, $1 \leqq j \leqq k$, VTX of a vertex v_{jq} ($1 \leqq q \leqq p_j$) is denoted by <u>VTX(v_{jq})</u>, DEG of a vertex v_{jq} is denoted by <u>DEG(v_{jq})</u>, BLG of a vertex v_{jq} is denoted by <u>BLG(v_{jq})</u> and LINK of a vertex v_{jq} is denoted by <u>LINK(v_{jq})</u>.

4.2 <u>Algorithm</u>

In this section, discuss the algorithm of solving the kC-problem. The algorithm is the following.

<u>Algorithm kGC.</u>
<u>Begin</u>
1. perform Procedure Prep ; <u>If</u> {status ≠ 0} <u>then</u> <u>go to</u> Step 5 ;
2. perform Procedure Settle ;
3. <u>while</u> {POINT(H_2) ≠ Λ} <u>do</u> <u>begin</u>
   u ← POINT(H_2) ; perform Procedure Edgadd ;
   <u>If</u> {status ≠ 0} <u>then</u> <u>go to</u> Step 5 ; perform Procedure Settle <u>end</u> ;
4. f : b_1, b_2, ⋯, b_p, is a k-partite graphical sequence and G is a k-partite graph with, for every j, $1 \leqq j \leqq p$, $\deg(u_j) = b_j$; halt ;

5. f : b_1, b_2, ····, b_p, is not a k-partite graphical sequence
End. (Algorithm kGC terminates.)

Procedure Prep.
Begin
1. status ← 0 ; POINT(H_1) ← Λ ; Make three sequences f, g_1 and g_2 ;
2. For j = 1, p − 1 do $POINT_n(j)$ ← Λ for every n, $1 \leqq n \leqq 2$;
For j = 1, k do $x_j \leftarrow \Sigma^{pj}_{q=1} d_{jq}$;
3. For j = 1, p do begin
If $\{b_j \geqq p\}$ then go to Step 8 end ;
4. $x \leftarrow \Sigma^{k}_{j=1} x_j$; $z_1 \leftarrow x - x_k$;
5. If {x is an odd number} or $\{x_k > z_1\}$ then go to Step 8 ;
6. G ← G = (V, E), where E = Φ ;
7. For j = 1, p do begin
If $\{b_j > 0\}$ then begin
LINK(u_j) ← $POINT_1(b_j)$; DEG(u_j) ← b_j ; VTX(u_j) ← u_j ; BLG(u_j) ← n_j ;
$POINT_1(b_j)$ ← VTX(u_j) end end ; halt ;
8. status ← 1 ;
End. (Procedure Prep terminates.)

Procedure Edgadd.
Begin
1. status ← 0 ;
2. while {POINT(H_1) ≠ Λ} do begin
v ← POINT(H_1) ; POINT(H_1) ← LINK(v) ; r ← DEG(v) ; LINK(v) ← $POINT_1(r)$;
$POINT_1(r)$ ← VTX(v) end ;
3. For j = p − 1, 1, -1 do begin
while {$POINT_1(j)$ ≠ Λ} do begin
v ← $POINT_1(j)$; r ← BLG(u) ; h ← BLG(v) ;
If {h = r} then j' ← j
else begin
$x' \leftarrow x - 2$; $x'_r \leftarrow x_r - 1$; $x'_h \leftarrow x_h - 1$;
z ← min(r, h) ; q ← max(r, h) ;
$y \leftarrow \max(x'_r, x'_h, x_1, \cdots, x_{z-1}, x_{z+1}, \cdots, x_{q-1}, x_{q+1}, \cdots, x_k)$;
If $\{y > x' - y\}$ then j' ← j
else begin
G ← G + e, where e = (VTX(u), VTX(v)) ; DEG(u) ← DEG(u) − 1 ;
DEG(v) ← DEG(v) − 1 ; $x \leftarrow x'$; $x_r \leftarrow x'_r$; $x_h \leftarrow x'_h$;
j' ← j − 1 end end ;
$POINT_1(j)$ ← LINK(v) ;
If {DEG(v) > 0} then begin
LINK(v) ← $POINT_2(j')$; $POINT_2(j')$ ← VTX(v) end ;
If {DEG(u) = 0} then go to Step 5 end end ;
4. If {DEG(u) > 0} then status ← 1 ;
5. halt

End. (Procedure Edgadd terminates.)

Procedure Settle.

Begin

1. For $j=1, p-1$ do begin

For $n=1, 2$ do begin

while $\{POINT_n(j) \neq \Lambda\}$ do begin

$v \leftarrow POINT_n(j)$; $POINT_n(j) \leftarrow LINK(v)$; $LINK(v) \leftarrow POINT(H_1)$;

$POINT(H_1) \leftarrow VTX(v)$ end end end ;

2. If $\{POINT(H_1) \neq \Lambda\}$ then begin

$v \leftarrow POINT(H_1)$; $POINT(H_1) \leftarrow LINK(v)$; $LINK(v) \leftarrow \Lambda$; $POINT(H_2) \leftarrow VTX(v)$

end

End. (Procedure Settle terminates.)

Example. Set $d_{11}=1$, $d_{12}=3$, $d_{21}=2$, $d_{22}=2$, $d_{23}=2$, $d_{31}=1$, $d_{32}=2$, $d_{33}=2$ and $d_{34}=3$. Then the following [1] through [4] are obtained.

[1] By Step 1 and 2, f : 1, 1, 2, 2, 2, 2, 2, 3, 3, g_1 : v_{31}, v_{11}, v_{32}, v_{33}, v_{21}, v_{22}, v_{23}, v_{34}, v_{12}, g_2 : 3, 1, 3, 3, 2, 2, 2, 3, 1, and the following data structure are obtained.

H_1 [→] → [v_{12}, 3, 1, →] → [v_{32}, 2, 3, →] → [v_{33}, 2, 3, →] → [v_{21}, 2, 2, →] →
→ [v_{22}, 2, 2, →] → [v_{23}, 2, 2, →] → [v_{31}, 1, 3, →] → [v_{11}, 1, 1, Λ]

H_2 [→] → [v_{34}, 3, 3, Λ]

[2] By Step 3, three edges (v_{34},v_{12}), (v_{34},v_{23}), (v_{34},v_{22}), and the following data structure are obtained.

LST_1 : 1 [→] → [v_{11}, 1, 1, →] → [v_{31}, 1, 3, Λ] , 3 [Λ] , ····, 8 [Λ]
2 [→] → [v_{21}, 2, 2, →] → [v_{33}, 2, 3, →] → [v_{32}, 2, 3, Λ]

LST_2 : 1 [→] → [v_{22}, 1, 2, →] → [v_{23}, 1, 2, Λ] , 2 [→] → [v_{12}, 2, 1, Λ] ,
3 [Λ] , ····, 8 [Λ]

[3] Similarly, the following edges are obtained.

(1) (v_{12},v_{21}), (v_{12},v_{33}) (2) (v_{32},v_{11}), (v_{32},v_{22})

(3) (v_{31},v_{23}) (4) (v_{21},v_{33})

[4] Hence a final graph being shown in Fig.4.1, can be obtained.

A final graph G is a 3-partite graph ($k=3$) and G satisfies $deg(v_{11})=1$, $deg(v_{12})=3$, $deg(v_{21})=2$, $deg(v_{22})=2$, $deg(v_{23})=2$, $deg(v_{31})=1$, $deg(v_{32})=2$, $deg(v_{33})=2$ and $deg(v_{34})=3$.

5. Time complexity

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

The time complexity of Procedure Prep is the following : Step 1 is $O(p \cdot \log_2 p)$, Step 4 is $O(k)$ and Step 2, 3, 5 and 7 are $O(|V|)$, where $|V|=p$. Thus the time complexity of Procedure Prep is $O(p \cdot \log_2 p)$.

The time complexity of Procedure Edgadd is the following : Step 2 is $O(|V|)$

and Step 3 is $O(k|V|)$. Thus the time complexity of Procedure Edgadd is $O(k\cdot|V|)$.

It is clear that the time complexity of Procedure Settle is $O(|V|)$.

The time complexity of Algorithm kGC is the following : Step 1 is $O(p\cdot\log_2 p)$, Step 2 is $O(|V|)$ and, Step 3 is $O(k|V|^2)$ since Procedure Edgadd and Settle are performed at $p-1$ times. Hence the time complexity of Algorithm kGC is $O(k|V|^2)$.

6. Conclusion

In this paper, a k-partite graph construction algorithm which performs the following (1) through (4), has been obtained :

(1) For a given sequence set $S=(s_1, s_2, \cdots, s_k)$, $s_j : d_{j1}, d_{j2}, \cdots, d_{j,pj}$, for every j, $1\leqq j\leqq k$, decide that whether $\Sigma^{k-1}{}_{j=1}x_j \geqq x_k$ is satisfied, where $x_j=\Sigma^{pj}{}_{q=1}d_{jq}$ for every j, $1\leqq j\leqq k$.

(2) If $\Sigma^{k-1}{}_{j=1}x_j\geqq x_k$ is satisfied then decide that whether a k-partite graph $G=(V_1\cup V_2\cup\cdots\cup V_k, E)$ such that $V_j=\{v_{j1}, v_{j2}, \cdots, v_{j,pj}\}$ for every j, $1\leqq j\leqq k$, and such that, for every q, $1\leqq q\leqq p_j$, $\deg(v_{jq})=d_{jq}$ holds, can be constructed from $S=(s_1, s_2, \cdots, s_k)$, and

(3) If such a k-partite graph can be constructed then $S=(s_1, s_2, \cdots, s_k)$ is a k-partite graphical sequence set.

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

I want to find an (approximation) algorithm of weighted version for further investigation.

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, Journal of the Society for Industrial and Applied Mathematics, 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-AL-27(1992), 31-38

[7] M.Takahashi, An Algorithm of Constructing a k-partite Multigraph from a k-

partite Multigraphical Sequence Set, Information Processing Society of Japan, Tech. Rep. 92-AL-28(1992), 17-24

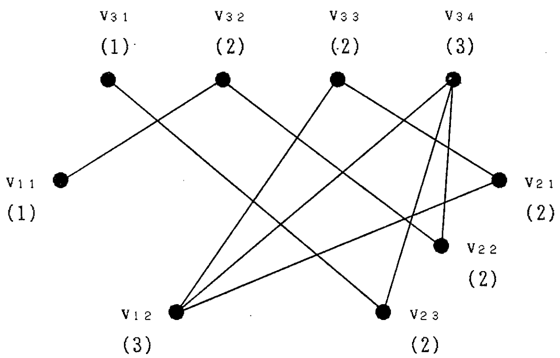

Fig.4.1