アルゴリズム 33−15
（1993．5．29）

グラフ次数列問題

高橋昌也 愛知技術短期大学電子工学科

今井桂子 浅野孝夫 中央大学理工学部情報工学科

非負整数の列 $S = (s_1, s_2, ..., s_n)$ がグラフ的であるとは、それを次数列としてもつようなグラフが存在することであり、グラフ次数列問題とは、与えられた非負整数列 $S = (s_1, s_2, ..., s_n)$ に対して、Sがグラフ的であるかどうかを判定し、もしそうならば、Sを次数列としてもつグラフを構成する問題である。本論文では、各種のグラフ次数列問題を考え、効率的アルゴリズムを与える。

Graphical Degree Sequence Problems

Masaya Takahashi
Department of Electronic Engineering, Aichi College of Technology

Keiko Imai and Takao Asano
Department of Information and System Engineering
Chuo University, Bunkyo-ku, Tokyo 112, Japan

A sequence of nonnegative integers $S = (s_1, s_2, ..., s_n)$ is *graphical* if there is a graph with vertices $v_1, v_2, ..., v_n$ such that $deg(v_i) = s_i$ for each $i = 1, 2, ..., n$. The graphical degree sequence problem is: Given a sequence S of nonnegative integers, determine whether it is graphical and, if so, construct a graph having S as a degree sequence. In this paper, we consider several variations of the graphical degree sequence problem and give efficient algorithms.

1 Introduction

A sequence of nonnegative integers $S = (s_1, s_2, ..., s_n)$ is *graphical* if there is a graph with vertices $v_1, v_2, ..., v_n$ such that $deg(v_i) = s_i$ for each $i = 1, 2, ..., n$ ($deg(v_i)$ is the degree of v_i). 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 [9] and then considered by Erdös and Gallai [5] and Hakimi [7].

Many variations can be considered. For example, if we admit multigraphs, then the multigraphical version is obtained. Takahashi [10] recently studied variations described below. A set of sequences of nonnegative integers $\{S_1, S_2, ..., S_k\}$ with $S_i = (s_{i1}, s_{i2}, ..., s_{in_i})$ is *k-partite graphical* (*k-partite multigraphical*) if there is a k-partite graph (k-partite multigraph) of k independent vertex sets $\{V_1, V_2, ..., V_k\}$ with $V_i = \{v_{i1}, v_{i2}, ..., v_{in_i}\}$ such that $deg(v_{ij_i}) = s_{ij_i}$ for each $i = 1, 2, ..., k$ and $j_i = 1, 2, ..., n_i$. 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 [10]. However, there were holes.

The problems stated above are all about undirected graphs. Directed versions can also be defined. For example, a pair of nonnegative integer sequences (S^+, S^-) with $S^+ = (s_1^+, s_2^+, ..., s_n^+)$ and $S^- = (s_1^-, s_2^-, ..., s_n^-)$ is *digraphical* (*multidigraphical*) if there is a directed graph (directed multigraph) with vertices $v_1, v_2, ..., v_n$ such that $deg^+(v_i) = s_i^+$ and $deg^-(v_i) = s_i^-$ for each $i = 1, 2, ..., n$ ($deg^+(v_i)$ and $deg^-(v_i)$ are the outdegree and indegree of v_i respectively). The digraphical (multidigraphical) degree sequence problem is defined similarly.

In this paper, we consider variations of the graphical degree sequence problem described above and present efficient algorithms including an $O(n)$ time algorithm to determine whether a sequence of nonnegative integers $S = (s_1, s_2, ..., s_n)$ is graphical or not and an $O(m)$ time algorithm to construct a graph having S as a degree sequence if S is graphical ($m = \sum_{i=1}^{n} s_i/2$).

2 Graphical Degree Sequence Problem

In this section, we consider the graphical degree sequence problem and present efficient algorithms. We first recall the previous results. Havel [9] and Hakimi [7] gave Proposition 1 independently and Erdös and Gallai [5] gave Proposition 2 in the following. Their proofs can be found in standard books of graph theory [3, 8].

Proposition 1. Let $S = (s_1, s_2, ..., s_n)$ be a sequence of nonnegative integers with $n > s_1 \geq s_2 \geq ... \geq s_n$ and let $S' = (s'_1, s'_2, ..., s'_{n-1})$ be a sequence of nonnegative integers obtained from S by setting $s'_i = s_{i+1} - 1$ ($i = 1, 2, ..., s_1$) and $s'_j = s_{j+1}$ ($j = s_1 + 1, ..., n - 1$). Then S is graphical if and only if S' is graphical.

Proposition 2. Let $S = (s_1, s_2, ..., s_n)$ be a sequence of nonnegative integers with $n > s_1 \geq s_2 \geq ... \geq s_n$. Then S is graphical if and only if $\sum_{k=1}^{i} s_k \leq i(i-1) + \sum_{k=i+1}^{n} \min\{i, s_k\}$ for each $i = 1, 2, ..., n$.

For a sequence of nonnegative integers $S = (s_1, s_2, ..., s_n)$ with $n > s_1 \geq s_2 \geq ... \geq s_n$, if $\sum_{i=1}^{n} s_i$ is odd then S is not graphical. Furthermore, if $s_n = 0$ then S is graphical if and only if $(s_1, s_2, ..., s_{n-1})$ is graphical. Thus, we assume throughout this section that $\sum_{i=1}^{n} s_i$ is even, $s_n \geq 1$ and $m = \sum_{i=1}^{n} s_i/2$.

Based on Proposition 1, one can easily determine whether S is graphical or not in $O(n^2)$ time. A graph G with S as a degree sequence can also be obtained in $O(n^2)$ time. Similarly, it is trivial to determine whether S is graphical or not in $O(n^2)$ time based on Proposition 2. However, it is difficult to use Proposition 2 to actually obtain a graph G with S as a degree sequence even if we know in advance that S is graphical.

In the following, we first present an algorithm to determine whether S is graphical or not in $O(n)$ time based on Proposition 2. Next we give an algorithm for actually constructing a graph G with S as a degree sequence in $O(m)$ time if we know in advance that S is graphical.

Let $S = (s_1, s_2, ..., s_n)$ be a given sequence of positive integers with $n > s_1 \geq s_2 \geq ... \geq s_n$. Let $a_i = \sum_{k=1}^{i} s_k$ and $b_i = \sum_{k=i+1}^{n} \min\{i, s_k\}$. Then Proposition 2 will be: S is graphical if and only if $a_i \leq i(i-1) + b_i$ for each $i = 1, 2, ..., n$.

Since $a_1 = s_1$ and $a_i = a_{i-1} + s_i$ for each $i = 2, 3, ..., n$, we can compute all a_i in $O(n)$ time. We will show that all b_i can also be computed in $O(n)$ time. Let $\sigma(i) = \min\{j | s_j < i\}$ for each $i = 1, 2, ..., n$ (we assume $s_{n+1} = 0$). Then σ is a decreasing function because $s_1 \geq s_2 \geq ... \geq s_n$, and thus we can compute, in $O(n)$ time, all $\sigma(i)$ and a unique integer α such that $\sigma(\alpha) > \alpha$ and $\sigma(\alpha+1) \leq \alpha+1$. If $i \geq \alpha+1$ then $\sigma(i) \leq \sigma(\alpha+1) \leq \alpha+1 \leq i$ and $s_k \leq s_i \leq s_{\sigma(i)} < i$ $(k = i+1, ..., n)$, and thus, $b_i = \sum_{k=i+1}^{n} s_k = a_n - a_i$. For each $i = 2, 3, ..., \alpha$, we have $\sigma(i) \geq \sigma(\alpha) > \alpha \geq i$ and $s_k \geq s_{\sigma(i)-1} \geq i$ $(k = i+1, ..., \sigma(i)-1)$ and $s_k \leq s_{\sigma(i)} < i$ $(k = \sigma(i), ..., n)$ and thus, $b_i = i(\sigma(i) - 1 - i) + a_n - a_{\sigma(i)-1}$. Of course, $b_1 = n - 1$. Thus all b_i can also be computed in $O(n)$ time. Since sorting n integers between 1 and n requires $O(n)$ time [1], we obtain the following theorem.

Theorem 1. For a sequence of nonnegative integers $S = (s_1, s_2, ..., s_n)$, we can determine whether S is graphical or not in $O(n)$ time.

Next we present an algorithm for actually constructing a graph for a given graphical degree sequence S based on Proposition 1. One drawback to use Proposition 1 is that $s'_1 \geq s'_2 \geq ... \geq s'_{n-1}$ does not always hold in S'. Thus we have to sort again to use Proposition 1 recursively. To avoid sorting, we modify the proposition as follows. It can be proved in the same way as Proposition 1.

Proposition 3. Let $S = (s_1, s_2, ..., s_n)$ be a sequence of positive integers with $n > s_1 \geq s_2 \geq ... \geq s_n$ and let $T = (t_1, t_2, ..., t_{n-1})$ be defined by using $k = s_n$, $x = \min\{j | s_j = s_k\}$ and $y = \max\{j | j \leq n-1, s_j = s_k\}$ as follows.

$$t_i = \begin{cases} s_i - 1 & \text{if } 1 \leq i \leq x-1 \text{ or } y-k+x \leq i \leq y, \\ s_i & \text{if } x \leq i \leq y-k+x-1 \text{ or } y+1 \leq i \leq n-1. \end{cases}$$

Then S is graphical if and only if T is graphical. Furthermore, $t_1 \geq t_2 \geq ... \geq t_{n-1}$.

Based on Proposition 3, we can obtain the following iterative algorithm CG for constructing a graph G having S as a degree sequence. In the algorithm, L is first set $L = \{j | s_{j-1} > s_j, j = 2, 3, ..., n\} \cup \{1\}$ and represented by a doubly-linked list and $pre[j] < j < suc[j]$ for each $j \in L$, where $pre[j]$ and $suc[j]$ denote the previous element and the next element of $j \in L$. Note that $T = (t_1, t_2, ..., t_n)$ is initialized $T = S$ and then maintained to satisfy $t_1 \geq t_2 \geq \cdots \geq t_n$. L is also maintained to satisfy $L = \{j | t_{j-1} > t_j\} \cup \{1\}$. Thus, $t_{pre[j]} = t_{pre[j]+1} = \cdots = t_{j-1} > t_j = t_{j+1} = \cdots = t_{suc[j]-1}$ for each $j \in L$.

Algorithm CG.
Input. A graphical sequence $S = (s_1, s_2, ..., s_n)$ with $n > s_1 \geq s_2 \geq ... \geq s_n \geq 1$.
procedure CG; {**comment** this calls procedure add_edge described below}
begin
  for $i := 1$ **to** n **do** $t_i := s_i$;
  $L := \{1\}$; **for** $i := 2$ **to** n **do if** $s_{i-1} > s_i$ **then** insert i into L;
  for $h := n$ **downto** 1 **do begin**
    if h is not in L **then** insert h into L; add_edge(h); delete h from L **end**
end;

```
procedure add_edge(h); { this adds edges between v_h and other vertices appropriately}
begin
    if t_h ≠ 0 then begin
        j :=the first element of L; while j ≤ t_h do j := suc[j]; {j > t_h ≥ pre[j]}
        for i := 1 to pre[j] − 1 do begin add edge (v_h, v_i); t_i := t_i − 1 end;
        j_new := j − t_h + pre[j] − 1;
        for i := j_new to j − 1 do begin add edge (v_h, v_i); t_i := t_i − 1 end;
        if t_h < j − 1 then begin {j_new − pre[j] = j − 1 − t_h > 0 and t_{j_new−1} > t_{j_new}}
            insert j_new into L; {j_new is inserted between pre[j] and j}
            if t_{pre[j]} = t_{pre[j]−1} then delete pre[j] from L end;
        if t_j = t_{j−1} then delete j from L
    end
end;
```

It is easy to see that Algorithm CG correctly constructs a graph G with S as a degree sequence and that it takes $O(m)$ time, if we observe that $pre[j]$ and $j-1$ play roles of x and y in Proposition 3 respectively. Thus we have the following theorem.

Theorem 2. For a graphical degree sequence $S = (s_1, s_2, ..., s_n)$, a graph with S as a degree sequence can be obtained in $O(m)$ time, where $m = \sum_{i=1}^{n} s_i/2$.

3 Bipartite Graphical Sequence Problem

In this section we consider the bipartite graphical degree sequence problem: Given a pair of nonnegative integer sequences $\{S_1, S_2\}$ with $S_1 = (s_{11}, s_{12}, ..., s_{1n_1})$ and $S_2 = (s_{21}, s_{22}, ..., s_{2n_2})$, determine whether $\{S_1, S_2\}$ is bipartite graphical (i.e., there is a bipartite graph with two independent vertex sets $V_1 = \{v_{11}, v_{12}, ..., v_{1n_1}\}$ and $V_2 = \{v_{21}, v_{22}, ..., v_{2n_2}\}$ such that $deg(v_{ij_i}) = s_{ij_i}$ for each $i = 1, 2$ and $j_i = 1, 2, ..., n_i$). The bipartite graphical degree sequence problem can be solved in almost the same way as the graphical degree sequence problem. We assume $\sum_{i=1}^{n_1} s_{1i} = \sum_{j=1}^{n_2} s_{2j}$ because otherwise $\{S_1, S_2\}$ is not bipartite graphical. We can also assume without loss of generality that $n_2 \geq s_{11} \geq s_{12} \geq \cdots \geq s_{1n_1}$ and $s_{21} \leq s_{22} \leq \cdots \leq s_{2n_2} \leq n_1$. Then the following proposition corresponding to Proposition 2 holds.

Proposition 4. Let $S_1 = (s_{11}, s_{12}, ..., s_{1n_1})$ and $S_2 = (s_{21}, s_{22}, ..., s_{2n_2})$ be a pair of nonnegative integer sequences with $\sum_{i=1}^{n_1} s_{1i} = \sum_{j=1}^{n_2} s_{2j}$, $n_2 \geq s_{11} \geq s_{12} \geq \cdots \geq s_{1n_1}$ and $s_{21} \leq s_{22} \leq \cdots \leq s_{2n_2} \leq n_1$. Then $\{S_1, S_2\}$ is bipartite graphical if and only if $\sum_{k=1}^{i} s_{1k} \leq i(n_2-j)+\sum_{k=1}^{j} s_{2k}$ for each $i = 1, 2, ..., n_1$, $j = 1, 2, ..., n_2$.

This proposition can be proved by the max-flow min-cut theorem by Ford and Fulkerson [6]. Based on Proposition 4, we can determine whether $\{S_1, S_2\}$ is bipartite graphical or not in $O(n)$ time ($n = n_1 + n_2$) as follows.

For each $i = 1, 2, ..., n_1$, we consider $\rho(i)$ defined by $\rho(i) = \max\{j | s_{2j} < i\}$ (we assume $s_{20} = 0$). Then Proposition 4 can be rewritten as follows: $\{S_1, S_2\}$ is bipartite graphical if and only if $\sum_{k=1}^{i} s_{1k} \leq i(n_2 - \rho(i)) + \sum_{k=1}^{\rho(i)} s_{2k}$ for each $i = 1, 2, ..., n_1$, since

$$i(n_2 - j) + \sum_{k=1}^{j} s_{2k} = in_2 + \sum_{k=1}^{j}(s_{2k} - i) \geq in_2 + \sum_{k=1}^{\rho(i)}(s_{2k} - i) = i(n_2 - \rho(i)) + \sum_{k=1}^{\rho(i)} s_{2k}$$

for each $j = 1, 2, ..., n_2$. We can compute all $\rho(i)$ in $O(n)$ time, since ρ is an increasing function and $s_{21} \leq s_{22} \leq \cdots \leq s_{2n_2}$. Thus, we have the following theorem.

Theorem 3. For a pair of nonnegative integer sequences $S_1 = (s_{11}, s_{12}, ..., s_{1n_1})$ and $S_2 = (s_{21}, s_{22}, ..., s_{2n_2})$, we can determine whether $\{S_1, S_2\}$ is bipartite graphical or not in $O(n)$ time, where $n = n_1 + n_2$.

Next we present an algorithm for actually constructing a bipartite graph for a given bipartite graphical set of degree sequences based on the following proposition, which can be proved easily.

Proposition 5. Let $S_1 = (s_{11}, s_{12}, ..., s_{1n_1})$ and $S_2 = (s_{21}, s_{22}, ..., s_{2n_2})$ be a pair of nonnegative integer sequences with $\sum_{i=1}^{n_1} s_{1i} = \sum_{j=1}^{n_2} s_{2j}$, $s_{11} \geq s_{12} \geq \cdots \geq s_{1n_1}$ and $s_{21} \leq n_1$. Let $T = (t_1, t_2, ..., t_{n_1})$ be defined by using $k = s_{21}$, $x = \min\{j | s_{1j} = s_{1k}\}$ and $y = \max\{j | s_{1j} = s_{1k}\}$ as follows.

$$t_i = \begin{cases} s_{1i} - 1 & \text{if } 1 \leq i \leq x-1 \text{ or } y-k+x \leq i \leq y, \\ s_{1i} & \text{if } x \leq i \leq y-k+x-1 \text{ or } y+1 \leq i \leq n_1. \end{cases}$$

Then $\{S_1, S_2\}$ is bipartite graphical if and only if $\{T, S_2 - s_{21}\}$ is bipartite graphical, where $S_2 - s_{21} = (s_{22}, s_{23}, ..., s_{2n_2})$. Furthermore, $t_1 \geq t_2 \geq ... \geq t_{n_1}$.

Based on Proposition 5, we can obtain the following algorithm CBG for constructing a bipartite graph G with $\{S_1, S_2\}$ as a pair of degree sequences. The algorithm is almost the same as Algorithm CG. In the algorithm, L is initialized $L = \{j | s_{1j-1} > s_{1j}, j = 1, 2, ..., n_1\} \cup \{1, n_1+1\}$ and represented by a doubly-linked list as before. $T = (t_1, t_2, ..., t_n)$ is initialized to be $T = S_1$ and then maintained to satisfy $t_1 \geq t_2 \geq \cdots \geq t_n$. L is also maintained to satisfy $L = \{j | t_{j-1} > t_j\}$. $pre[j]$ and $j-1$ play roles of x and y in Proposition 5.

Algorithm CBG.
Input. A bipartite graphical set of degree sequences $\{S_1, S_2\}$ with $S_i = (s_{i1}, s_{i2}, ..., s_{in_i})$ $(i = 1, 2)$ and $n_2 \geq s_{11} \geq s_{12} \geq \cdots \geq s_{1n_1}$.

```
procedure CBG;
begin
    for i := 1 to n1 do t_i := s_1i;
    L := {1, n1 + 1}; for i := 2 to n1 do if s_1i-1 > s_1i then insert i into L;
    for  h := 1 to n2 do
        if s_2h ≠ 0 then begin
            j :=the first element of L; while j ≤ s_2h do j := suc[j]; {j > s_2h ≥ pre[j]}
            for i := 1 to pre[j] − 1 do begin add edge (v_2h, v_1i); t_i := t_i − 1 end;
            j_new := j − s_2h + pre[j] − 1;
            for i := j_new to j − 1 do begin add edge (v_2h, v_1i); t_i := t_i − 1 end;
            if s_2h < j − 1 then begin
                insert j_new into L; if t_pre[j] = t_pre[j]−1 then delete pre[j] from L end;
            if t_j = t_j−1 then delete j from L
        end
end;
```

It is easy to see that Algorithm CBG correctly constructs a bipartite graph G with $V_i = \{v_{i1}, v_{i2}, ..., v_{in_i}\}$ $(i = 1, 2)$ having $\{S_1, S_2\}$ as a pair of degree sequences and that it takes $O(m)$ time, where $m = \sum_{j=1}^{n_1} s_{1j}$. Thus we have the following theorem.

Theorem 4. For a bipartite graphical set of degree sequences $\{S_1, S_2\}$ with $S_i = (s_{i1}, s_{i2}, ..., s_{in})$ $(i = 1, 2)$, a bipartite graph G with $V_i = \{v_{i1}, v_{i2}, ..., v_{in_i}\}$ $(i = 1, 2)$ having $\{S_1, S_2\}$ as a pair of degree sequences can be obtained in $O(m)$ time, where $m = \sum_{j=1}^{n_1} s_{1j}$.

We note that the digraphical degree sequence problem shrares many properties in common with the bipartite graphical degree sequence problem here and can be solved in almost the same way.

4 Multigraphical Degree Sequence Problem

In this section we consider multigraphical versions of the degree sequence problem. We first consider the *multigraphical degree sequence problem*: Given a sequence of nonnegative integers $S = (s_1, s_2, ..., s_n)$, determine whether it is *multigraphical* (that is, there is a graph with vertices $v_1, v_2, ..., v_n$ such that $deg(v_i) = s_i$ for each $i = 1, 2, ..., n$) or not. The following proposition can be easily obtained.

Proposition 6. Let $S = (s_1, s_2, ..., s_n)$ be a sequence of nonnegative integers such that $s_1 \geq s_2 \geq ... \geq s_n$ and $\sum_{i=1}^{n} s_i$ is even. Then S is multigraphical if and only if $s_1 \leq \sum_{i=2}^{n} s_i$.

Proof. Since the necessity is trivial, we consider only the sufficiency. We use induction on n. If $n = 2$ then $s_1 = s_2$ and thus S is multigraphical. If $n = 3$ then there is a multigraph G with $(s_1 + s_2 - s_3)/2$ edges connecting v_1 and v_2, $(s_2 + s_3 - s_1)/2$ edges connecting v_2 and v_3 and $(s_3 + s_1 - s_2)/2$ edges connecting v_3 and v_1. Thus S is multigraphical.

Suppose the proposition is true for $n < N$ for an integer $N \geq 4$. Now consider $n = N$. Let $\Delta = s_1 - s_2$. We consider three cases.

Case 1. If $\Delta \geq s_n$ then $S' = (s_1 - s_n, s_2, ..., s_{n-1})$ satisfies the conditions $s_1 - s_n \geq s_2$ and $s_1 - s_n \leq \sum_{i=2}^{n-1} s_i$ and thus S' is multigraphical by the inductive hypothesis. Let G' be a multigraph having S' as a degree sequence. Then the graph obtained from G' by adding vertex v_n and s_n edges connecting v_1 and v_n has S as a degree sequence.

Case 2. If $0 < \Delta < s_n$, then $S' = (s_1 - \Delta, s_2, ..., s_{n-1}, s_n - \Delta)$ satsifies the conditions $s_1 - \Delta \geq s_2$ and $s_1 - \Delta \leq \sum_{i=2}^{n} s_i - \Delta$ and can be reduced to the following case since $s'_1 = s_1 - \Delta = s_2$.

Case 3. If $\Delta = 0$ then $S' = (s_1, s_2, ..., s_{n-2}, s_{n-1} - s_n)$ satisfies the conditions $s_1 = s_2$ and $s_1 \leq \sum_{i=2}^{n-1} s_i - s_n$ and S' is multigraphical by the inductive hypothesis. Thus S is multigraphical in Cases 2 and 3.

Based on Proposition 6 and its proof described above, one can easily obtain an $O(n)$ time algorithm to determine whether S is multigraphical and, if so, construct a multigraph G with S as a degree sequence. Note that, in the algorithm for constructing G, first Case 1 occurs (several times) and next Case 2 occurs (at most once) and then Case 3 occurs (several times) and finaly only three elements will remain and edges are added among the three vertices. Thus only $O(n)$ pairs are connected by edges.

Theorem 5. For a sequence of nonnegative integers $S = (s_1, s_2, ..., s_n)$, it can be determined in $O(n)$ time whether S is multigraphical or not. Furtheremore, if S is multigraphical and $s_1 \geq s_2 \geq ... \geq s_n$, then a multigraph G with S as a degree sequence can be constructed in $O(n)$ time.

Next we consider the bipartite multigraphical degree sequence problem: Given a pair of nonnegative integer sequences $\{S_1, S_2\}$ with $S_1 = (s_{11}, s_{12}, ..., s_{1n_1})$ and $S_2 = (s_{21}, s_{22}, ..., s_{2n_2})$, determine whether $\{S_1, S_2\}$ is bipartite multigraphical (that is, there is a bipartite multigraph with two independent vertex sets $V_1 = \{v_{11}, v_{12}, ..., v_{1n_1}\}$ and $V_2 = \{v_{21}, v_{22}, ..., v_{2n_2}\}$ such that $deg(v_{ij_i}) = s_{ij_i}$ for each $i = 1, 2$ and $j_i = 1, 2, ..., n_i$) or not. The bipartite multigraphical degree sequence problem can be solved easily.

Proposition 7. Let $S_1 = (s_{11}, s_{12}, ..., s_{1n_1})$ and $S_2 = (s_{21}, s_{22}, ..., s_{2n_2})$ be a pair of nonnegative integer sequences. Then $\{S_1, S_2\}$ is bipartite multigraphical if and only if $\sum_{i=1}^{n_1} s_{1i} = \sum_{j=1}^{n_2} s_{2j}$.

Based on Proposition 7, the following bipartite multigraph construction algorithm is immediately obtained: If $\{S_1, S_2\}$ is bipartite multigraphical, then add $\min\{s_{11}, s_{21}\}$ edges between two vertices v_{11}, v_{21} and do recursively for $\{S'_1, S'_2\}$, where

$$S_1' = \begin{cases} (s_{11} - s_{21}, s_{12}, ..., s_{1n_1}) & \text{if } s_{11} > s_{21} \\ (s_{12}, ..., s_{1n_1}) & \text{otherwise} \end{cases}$$

$$S_2' = \begin{cases} (s_{21} - s_{11}, s_{22}, ..., s_{2n_2}) & \text{if } s_{11} < s_{21} \\ (s_{22}, ..., s_{2n_2}) & \text{otherwise.} \end{cases}$$

Theorem 6. For a pair of nonnegative integer sequences $S_1 = (s_{11}, s_{12}, ..., s_{1n_1})$ and $S_2 = (s_{21}, s_{22}, ..., s_{2n_2})$, it can be determined in $O(n)$ time whether $\{S_1, S_2\}$ is bipartite multigraphical and, if so, a bipartite multigraph G with $\{S_1, S_2\}$ as a pair of degree sequences can be constructed in $O(n)$ time ($n = n_1 + n_2$).

Next we consider the k-partite multigraphical degree sequence problem: Given a set of sequences of nonnegative integers $\{S_1, S_2, ..., S_k\}$ with $S_i = (s_{i1}, s_{i2}, ..., s_{in_i})$, determine whether $\{S_1, S_2, ..., S_k\}$ is k-partite multigraphical (that is, there is a k-partite multigraph of k independent vertex sets $\{V_1, V_2, ..., V_k\}$ with $V_i = \{v_{i1}, v_{i2}, ..., v_{in_i}\}$ such that $deg(v_{ij_i}) = s_{ij_i}$ for each $i = 1, 2, ..., k$ and $j_i = 1, 2, ..., n_i$). The k-partite multigraphical degree sequence problem can also be solved easily.

Proposition 8. Let $\{S_1, S_2, ..., S_k\}$ be a set of sequences of nonnegative integers with $S_i = (s_{i1}, s_{i2}, ..., s_{in_i})$. Let $t_i = \sum_{j=1}^{n_i} s_{ij}$. Then $\{S_1, S_2, ..., S_k\}$ is k-partite multigraphical if and only if $\sum_{i=1}^{k} t_i$ is even and $\max_{i=1}^{k}\{t_i\} \le \sum_{i=1}^{k} t_i/2$.

Based on Proposition 8, it is trivial to obtain $O(n)$ time algorithm to determine whether $\{S_1, S_2, ..., S_k\}$ is k-partite multigraphical or not, where $n = n_1 + n_2 + \cdots + n_k$. To construct a k-partite graph G with $\{S_1, S_2, ..., S_k\}$ as a set of degree sequences, we first construct a condensed multigraph H with k vertices $w_1, w_2, ..., w_k$ which has $T = (t_1, t_2, ..., t_k)$ as a degree sequence. H can be obtained in $O(k)$ time based on the proof of Proposition 6, if we have done sorting of $\{t_1, t_2, ..., t_k\}$ in $O(k \log k)$ time. Suppose that there are just h_{ij} edges in H connecting two vertices w_i and w_j. Restricting the vertex set of G to V_i and V_j, we construct a bipartite multigraph G_{ij} with h_{ij} edges based on the bipartite multigraph construction algorithm described above and then modify the degree sequences of S_i and S_j. Repeating this procedure iteratively, we can construct a required multigraph G in $O(n)$ time.

Theorem 7. For a set of sequences of nonnegative integers $\{S_1, S_2, ..., S_k\}$ with $S_i = (s_{i1}, s_{i2}, ..., s_{in_i})$, it can be determined in $O(n)$ time whether $\{S_1, S_2, ..., S_k\}$ is k-partite multigraphical ($n = n_1 + n_2 + \cdots + n_k$) and, if so, a k-partite multigraph G with $\{S_1, S_2, ..., S_k\}$ as a set of degree sequences can be constructed in $O(n)$ time except for $O(k \log k)$ time for sorting.

Next we consider the multidigraphical degree sequence problem: Given a pair of nonnegative integer sequences (S^+, S^-) with $S^+ = (s_1^+, s_2^+, ..., s_n^+)$ and $S^- = (s_1^-, s_2^-, ..., s_n^-)$, determine whether it is *multidigraphical* (that is, there is a directed multigraph with vertices $v_1, v_2, ..., v_n$ such that $deg^+(v_i) = s_i^+$ and $deg^-(v_i) = s_i^-$ for each $i = 1, 2, ..., n$). The multidigraphical degree sequence problem can be solved in a similar way as the bipartite multigraphical degree sequence problem. The following theorem can be obtained by the max-flow min-cut theorem.

Proposition 9. Let (S^+, S^-) be a pair of nonnegative integer sequences with $S^+ = (s_1^+, s_2^+, ..., s_n^+)$ and $S^- = (s_1^-, s_2^-, ..., s_n^-)$. Then (S^+, S^-) is multidigraphical if and only if $\sum_{j=1}^{n} s_j^+ = \sum_{j=1}^{n} s_j^-$ and $s_i^+ + s_i^- \le \sum_{j=1}^{n} s_j^+$.

Based on Proposition 9, the multidigraphical degree sequence problem can be solved in $O(n)$ time. Furtheremore, if (S^+, S^-) is multigraphical, then a directed multigraph G with (S^+, S^-)

as a pair of degree sequences can be constructed in $O(n)$ time by Dinic's max-flow algorithm [4]. Thus, we have the following theorem.

Theorem 8. For a pair of nonnegative integer sequences $S^+ = (s_1^+, s_2^+, ..., s_n^+)$ and $S^- = (s_1^-, s_2^-, ..., s_n^-)$, it can be determined in $O(n)$ time whether (S^+, S^-) is multidigraphical and, if so, a directed multigraph G with (S^+, S^-) as a pair of degree sequences can be constructed in $O(n)$ time.

5 Concluding Remarks

We have considered variations of the graphical degree sequence problem and given optimal algorithms. Algorithms described here constructing a graph are all concerned with representing a graph explicitly. If an implicitly represented graph is satisfactory, for example, if we represent edges connecting vertex v_i and other vertices in terms of a constant number of intervals, we can obtain faster $O(n \log\log n)$ time algorithms for constructing graphs and bipartite graphs (see [2]). We conjecture that there may be $O(n)$ time for these problems.

Takahashi [10] considered the k-partite graphical problem. Although this problem is polynomially solvable based on maximum matching algorithms, Takahashi's algorithm is more efficient. However, we will give a remark here that there is a hole in his proof and his algorithm works only for 3-partite graphs.

Acknowledgments

The third author was supported in part by Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan, by the Ohkawa Institute of Information and Telecommunication, and by the Alexander von Humboldt Foundation.

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[2] T. Asano, *An $O(n \log\log n)$ time algorithm for constructing a graph with a prescibed degree sequence*, in preparation.

[3] M. Behzard, G. Chartrand, L.L.- Foster, *Graphs & Digraphs*, Prindle, Weber & Schmidt, 1979.

[4] E.A. Dinic, *Algorithm for solution of a problem of maximum flow in a network with power estimation*, Soviet Mathematics Doklady 11 (1970), 1277-1280.

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

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

[7] S. 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.

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

[9] V. Havel, *A remark on the existence of finite graphs (Hungarian)*, Časopis Pěst. Mat. 80 (1955), 477-480.

[10] 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.