An Algorithm for Graph Isomorphism

An Algorithm for Graph Isomorphism

Atsuro SENO*, Kunio FUKUNAGA** and Tamotsu KASAI**

(Received November 15, 1977)

The present paper describes an algorithm for finding efficiently the isomorphic graphs by using a new method which represents some given graphs in a Euclidean space named graphic space. The graph isomorphism is determined on the basis of a vertex-vector and a distance between two graphs defined in the graphic space.

It is shown that the processing time of the algorithm proposed in this paper is given as a function of the number of vertices, n, in the graph and is generally proportional to $n^4(\log n)^2$.

1. Introduction

The problem of determining whether two given graphs are isomorphic or not is important in its applications to many fields as well as study of property between the graphs.

The isomorphism of two labeled graphs with n vertices can be determined easily by the comparisons of n times between the corresponding labeled vertices of the graphs. In the case where the given two graphs have no labeled vertices, however, the determination of isomorphism must be done by the comparisons of $n!$ times between n vertices of the graphs. Therefore, the time required to determine the isomorphism of the graphs is proportional to $n!$ and the larger the value of n is made, the more rapidly the processing time is increased. From this fact, if the value of n is larger than 10, it can be seen that the determination of the graph isomorphism becomes almost impossible.

Many methods for decreasing the number of comparisons between the vertices of two graphs have been discussed and the efficient procedures for the graphs which have no particular subgraph have been devised.[1),2)] However, it is difficult to solve generally the problem by applying these procedures to general graphs.

There are a few procedure[3)~5)] for obtaining the solution by comparing the eigenvalues of adjacency matrices representing the relation of connection in each of the given graphs. However, since these procedures depend on only comparison between the eigenvalues of adjacency matrices, they leave something to be desired and there are some scopes for discussion.

In this paper, first, a new method which represents a graph in an n dimensional Euclidean space is introduced.[6),7)] Scuh a space is called a graphic space and the

* Graduate Student, Department of Electrical Engineering, College of Engineering.
** Department of Electrical Engineering, College of Engineering.

representation of the graph in this space is done on the basis of the weights associated with each edge of the graph. Next, an efficient algorithm to determine the graph isomorphism by using this representing method is described.

This algorithm has an excellent feature that there is no need to consider many comparisons between the vertices even if in the case where the given graphs have no labeled one, and it is possible to decrease the processing time required to determine the graph isomorphism. It is shown that the processing time by this algorithm is proportional to $n^4(\log n)^2$ if the given graphs are represented with a proper accuracy in the graphic space.

Finally, in order to make sure of efficiency of the algorithm described in this paper, a result applying it to two graphs with 50 vertices is shown as an example.

2. Graphic Space

Let G be a given connected and nondirected graph with m edges e_i $(i = 1, 2, \cdots, m)$ and n vertices v_j $(j = 1, 2, \cdots, n)$, and also let d_i $(i = 1, 2, \cdots, m)$ be the weights associated with each of m edges. In this paper, each weight is regard as the strength of combination between both end vertices of the edge.

We now consider a graphical representation of the relation between edges and vertices in a graph. This is done on basis of "Theory and Examples of Quantification (II)"[8] and is given by a joint distribution diagram taking vertices on the quardrature axis and edges on the longitudinal axis in the plane of the second order. The correlation coefficient ρ between edges and vertices on the joint distribution diagram is defined and the proper numerical values are given to each edge and vertex so as to make the maximum value of ρ.

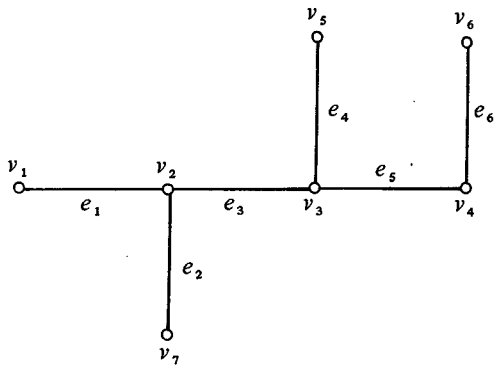

Fig. 1. An example of graph G.

For example, we consider a graph G as shown in Fig. 1. If a mark "o" is used to show the relation of connection between edges and vertices, the joint distribution diagram as shown in Fig. 2 is obtained.

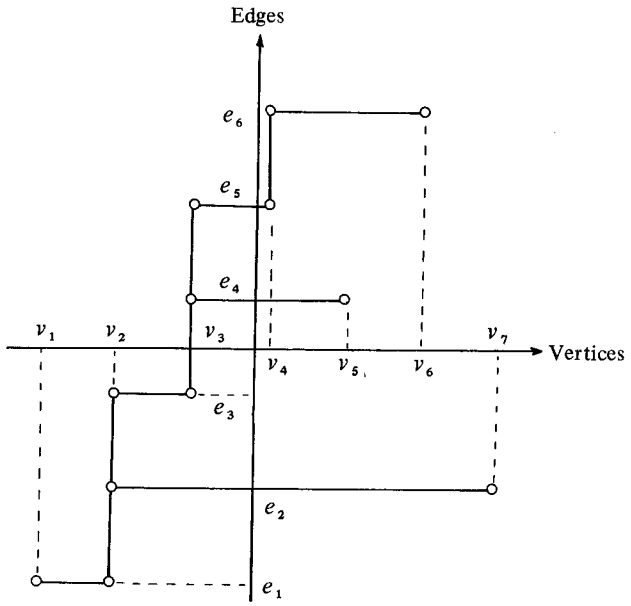

Fig. 2. A joint distribution diagram for graph G of Fig. 1.

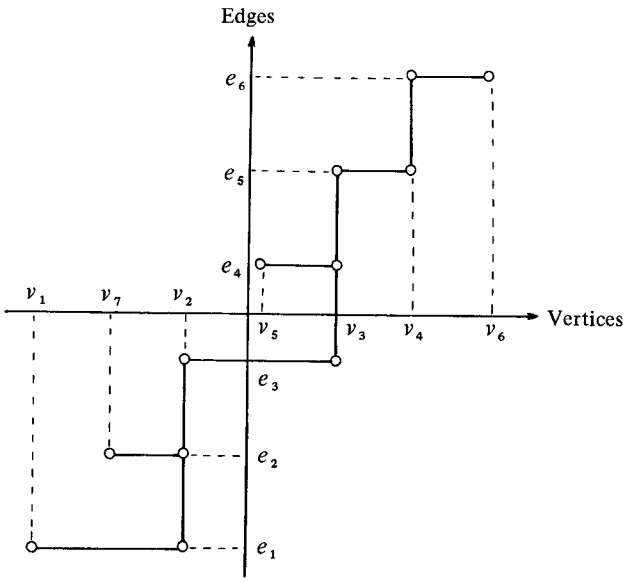

Fig. 3. The joint distribution diagram with maximum correlation coefficient.

By the way, the maximum value of correlation coefficient ρ is obtained when the marks "○" distribute along a straight line passing through the origin of coordinate in Fig. 2. According to this consideration, the proper numerical values are given to each of edges and vertices, and the joint distribution diagram of Fig. 2 is rearranged. Thus, the resulting joint distribution diagram becomes as shown in Fig. 3. In this figure, the neighboring

vertices are combined strongly each other and the distant vertices weakly. Therefore, it follows from this that the vertices are located in proportion to the weights of edges by maximizing the correlation coefficient.

In order to calculate practically the correlation coefficient, let x_j $(j = 1, 2, \cdots, n)$ be the variables showing the positions of the vertices v_j $(j = 1, 2, \cdots, n)$ on the X-axis and y_i $(i = 1, 2, \cdots, m)$ be the variables showing the positions of edges e_i $(i = 1, 2, \cdots, m)$ on the Y-axis. Then, it is supposed that the joint distribution diagram gives the probability distribution of two variables by considering the weights of edges:

$$P(X = x_j, \quad Y = y_i) = \delta_{ji} d_i / T_0 , \tag{1}$$

where

$$\delta_{ji} = \begin{cases} 1 & \text{if edge } e_i \text{ is incident at vertex } v_j, \\ 0 & \text{otherwise}, \end{cases}$$

and

$$T_0 = \sum_{j=1}^{n} \sum_{i=1}^{m} \delta_{ji} d_i . \tag{2}$$

If we take Eq. (1) as the probability density function $P(x_j, y_i)$, two probability functions for X and Y, $P(x_j)$ and $P(y_i)$, are

$$P_1(x_j) = \sum_{i=1}^{m} P(x_j, y_i), \tag{3}$$

$$P_2(y_i) = \sum_{j=1}^{n} P(x_j, y_i). \tag{4}$$

Furthermore, from the definition, the variances of X and Y, ${\sigma_X}^2$ and ${\sigma_Y}^2$, and the covariance of X and Y, C_{XY}, are represented, respectively, as follows:

$${\sigma_X}^2 = [\sum_{j=1}^{n} x_j^2 P_1(x_j)] - [\sum_{j=1}^{n} x_j P_1(x_j)]^2 , \tag{5}$$

$${\sigma_Y}^2 = [\sum_{i=1}^{m} y_i^2 P_2(y_i)] - [\sum_{i=1}^{m} y_i P_2(y_i)]^2 , \tag{6}$$

$$C_{XY} = \sum_{j=1}^{n} \sum_{i=1}^{m} x_j y_i P(x_j, y_i) - [\sum_{j=1}^{n} x_j P_1(x_j)] [\sum_{i=1}^{m} y_i P_2(y_i)] . \tag{7}$$

From Eqs. (5) ~ (7), the correlation coefficient ρ is given by

$$\rho = \frac{C_{XY}}{\sigma_X \sigma_Y} . \tag{8}$$

For x_j and y_i, the conditions to obtain a maximum value of ρ,

$$\frac{\partial \rho}{\partial x_j} = 0 , \qquad (j = 1, 2, \cdots, n) \tag{9}$$

and

$$\frac{\partial \rho}{\partial y_i} = 0 , \qquad (i = 1, 2, \cdots, m) \tag{10}$$

lead to

$$AX = \lambda CX , \quad ^{8)} \tag{11}$$

where $A = [\, a_{jk} \,]$ is an $n \times n$ symmetric matrix, $C = [\, c_{kk} \,]$ is a diagonal matrix and X is a column vector which is given by

$$X = (x_1, x_2, \cdots, x_n)^t , \tag{12}$$

where the superscript t means "transpose of" the matrix and the elements x_j $(j = 1, 2, \cdots, n)$ are the variables showing the position of the vertices v_j $(j = 1, 2, \cdots, n)$ on the X-axis in the joint distribution diagram. The entries a_{jk} and c_{kk} of the matrices A and C are

$$a_{jk} = \frac{1}{2} \sum_{i=1}^{m} \delta_{ki} \delta_{ji} d_i , \tag{13}$$

$$c_{kk} = \sum_{i=1}^{m} \delta_{ki} d_i = 2a_{kk} , \tag{14}$$

and the parameter λ is given by

$$\lambda = \rho^2 . \tag{15}$$

Now let $\lambda_1, \lambda_2, \cdots, \lambda_n$ $(1 \geqq \lambda_1 \geqq \lambda_2 \geqq \cdots \geqq \lambda_n \geqq 0)$ be the eigenvalues obtaining by solving Eq. (11), and also let $X^{(k)}$ be the eigenvector associated with eigenvalues λ_k $(k = 1, 2, \cdots, n)$. Then, $X^{(k)}$ is

$$X^{(k)} = (x_{1k}, x_{2k}, \cdots, x_{nk})^t , \tag{16}$$

where the maximum absolute value of the elements is normalized to 1.

By the way, the greater the correlation coefficient ρ is made, the better the relation between edges and vertices is represented. Therefore, the accuracy of the representation in the Euclidean space depends on the value of ρ. As is evident from Eq. (11), if the weights of the edges in one of two given graphs are proportional to those in the other, the eigenvectors of the two graphs have the same elements.

In order to make correlation for the matter as just mentioned above, we now define a vector named vertex-vector. This is made from the eigenvector by using the average value d_0 of the weights d_i $(i = 1, 2, \cdots, m)$ and the correlation coefficient ρ.

[Definition 1] (Vertex-vector)

The *vertex-vector* X_j $(j = 1, 2, \cdots, n)$ for the vertex v_j $(j = 1, 2, \cdots, n)$ in a graph G are defined as follows:

$$X_j = (\sqrt{\lambda_1}\, x_{j2}, \sqrt{\lambda_2}\, x_{j2}, \cdots, \sqrt{\lambda_r}\, x_{jr})/d_0 , \tag{17}$$

where r $(\leqq n)$ is the number of dimensions of the vertex-vector, which relates to the accuracy of representation of the graph in a Euclidean space, and x_{jk} $(k = 1, 2, \cdots, r)$ has been defined in Eq. (16).

If we take the value of the vertex-vector in the coordinates as the position of the vertex, the graph G can be represented in a Euclidean space of r dimensions. In the following, we call the space where the graph is represented in such a method "*graphic space*".

In the representing method of the graph in the graphic space, each vertex-vector is obtained on the basis of the weights of edges. Consequently, if a given graph resembles another given one in the construction of edges and vertices or in the weights of edges, the arrangements of the vertices in the graphic space resemble each other, and if the graphs are different in the construction of edges and vertices, the arrangements of the vertices in the graphic space are also different. Therefore, when two given graphs are represented in the same graphic space, it can be considered that they are very similar if their vertices are arranged likewise, and not otherwise. From this, we now define a distance between two graphs as follows:

[Definition 2] (Distance between two graphs)

Consider two given graphs G_a and G_b with n vertices. Let $[v_{a1}, v_{a2}, \cdots, v_{an}]$ be the set of vertices in G_a and $[v_{b1}, v_{b2}, \cdots, v_{bn}]$ be the set of vertices in G_b corresponding to v_{aj} $(j = 1, 2, \cdots, n)$, and also let X_{aj} and X_{bj} be the vertex-vectors for v_{aj} and v_{bj}, respectively. Then the *distance between* G_a *and* G_b, $D(G_a, G_b)$, is defined as

$$D(G_a, G_b) = \sqrt{\frac{1}{r} \sum_{j=1}^{n} \| X_{aj} - X_{bj} \|^2} , \tag{18}$$

where r is the number of dimensions of the vertex-vectors X_{aj} and X_{bj}.

Thus, by this definition we have the following theorems:

[Theorem 1] The distance between the graphs satisfies the pseudo-distance. In other words, the distance on any given graphs G_a, G_b and G_c whose vertices correspond with one another satisfies the following conditions:

$$(1) \quad D(G_a, G_b) = 0 , \tag{19}$$

$$(2) \quad D(G_a, G_b) = D(G_b, G_a) , \tag{20}$$

$$(3) \quad D(G_a, G_b) + D(G_b, G_c) \geqq D(G_a, G_c). \tag{21}$$

(Proof) It is clear from the definition that the conditions (1) and (2) are satisfied. Let v_{bj} and v_{cj} $(j = 1, 2, \cdots, n)$ be the vertices in G_b and G_c corresponding to the vertices v_{aj} $(j = 1, 2, \cdots, n)$, and also let X_{aj}, X_{bj} and X_{cj} be the vertex-vectors

for the vertices v_{aj}, v_{bj} and v_{cj}. Then, from the property of a Euclidean space of r dimensions, it is clear that

$$\begin{aligned}
&\| X_{a1} - X_{b1} \| + \| X_{b1} - X_{c1} \| \geqq \| X_{a1} - X_{c1} \| , \\
&\| X_{a2} - X_{b2} \| + \| X_{b2} - X_{c2} \| \geqq \| X_{a2} - X_{c2} \| , \\
&\quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot
\end{aligned}$$

and

$$\| X_{an} - X_{bn} \| + \| X_{bn} - X_{cn} \| \geqq \| X_{an} - X_{cn} \| .$$

These equations lead to

$$\sqrt{\sum_{j=1}^{n} (\| X_{aj} - X_{bj} \| + \| X_{bj} - X_{cj} \|)^2} \geqq \sqrt{\sum_{j=1}^{n} \| X_{aj} - X_{cj} \|^2} .$$

Since it is generally satisfied that

$$\sqrt{\sum_{j=1}^{n} \| X_{aj} - X_{bj} \|^2} + \sqrt{\sum_{j=1}^{n} \| X_{bj} - X_{cj} \|^2} \geqq \sqrt{\sum_{j=1}^{n} (\| X_{aj} - X_{bj} \| + \| X_{bj} - X_{cj} \|)^2} ,$$

we have

$$\sqrt{\sum_{j=1}^{n} \| X_{aj} - X_{bj} \|^2} + \sqrt{\sum_{j=1}^{n} \| X_{bj} - X_{cj} \|^2} \geqq \sqrt{\sum_{j=1}^{n} \| X_{aj} - X_{cj} \|^2} .$$

Using the distance between the graphs defined just before, we obtain

$$D(G_a, G_b) + D(G_b, G_c) \geqq D(G_a, G_c) . \qquad \text{QED.}$$

[Theorem 2] The distance between two graphs G_a and G_b, $D(G_a, G_b)$, equals to 0, when they are isomorphic. That is, the necessary condition for graph isomorphism is

$$D(G_a, G_b) = 0 . \tag{22}$$

3. Algorithm for Graph Isomorphism

The representation of the graphs in the graphic space described in Chapter 2 is obtained by maximizing the correlation coefficient between edges and vertices based on the weights of edges. By this procedure, a given nondirected graph can be represented uniquely in the graphic space. Therefore, if two given graphs are represented in the same graphic space and the positions of the vertices in a graph are compared with those in the other, we can determine whether the graphs are isomorphic or not. We now give the algorithm for determining isomorphism of two nondirected graphs G_a and G_b with n vertices by the representing method of the graphs in the graphic space.

Step 1. Let $[v_{a1}, v_{a2}, \cdots, v_{an}]$ and $[v_{b1}, v_{b2}, \cdots, v_{bn}]$ be the vertices in

the given graphs G_a and G_b, respectively. Determine the minimum number of dimensions of the graphic space, r, required to represent separately the vertices in each graph and represent all vertices in the graphic space of r dimensions. Then let $[X_{a1}, X_{a2}, \cdots, X_{an}]$ and $[X_{b1}, X_{b2}, \cdots, X_{bn}]$ be the vertex-vectors for G_a and G_b whose eigenvalues do not overlap each other.

Step 2. Make a correspondence between the vertex-vectors for the vertices in G_a and G_b, which locate at the nearest distance, in the graphic space of r dimensions. If the vertex-vectors for the vertices in G_a and G_b have one to one correspondence, go to step 3. In contrast with this, if they have no one to one correspondence, the graphs G_a and G_b are not isomorphic. Stop.

Step 3. Suppose the vertex-vectors X_{aj} and X_{bj} $(j = 1, 2, \cdots, n)$ have one to one correspondence. If the distance of graphs, $D(G_a, G_b)$, is zero, go to step 4. If the distance is not zero, the graphs G_a and G_b are not isomorphic. Stop.

Step 4. Arrange the rows and the columns corresponding to v_{bj} in the adjacency matrix of G_b in the same order as those corresponding to v_{aj} in the adjacency matrix of G_a. After this arrangement, if and only if the adjacency matrices of G_a and G_b are the same, two graphs G_a and G_b are isomorphic. If not, they are not isomorphic.

4. Processing Time of Algorithm

In this chapter, the processing time of the algorithm described in Chapter 3 is evaluated by calculating the number of products and comparisons of the weights of edges.

First, each eigenvalue and eigenvector in the eigen equation (11) are obtained by using "*Power Method*" in step 1.

Let $\lambda_1, \lambda_2, \cdots, \lambda_n$ $(|\lambda_1| \geqq |\lambda_2| \geqq \cdots \geqq |\lambda_n|)$ be the eigenvalues of an $n \times n$ matrix A. Then any n dimensional vector x can be represented by the linear combination of the eigenvectors u_i $(i = 1, 2, \cdots, n)$ as

$$x = \sum_{i=1}^{n} c_i u_i , \tag{23}$$

where c_i $(i = 1, 2, \cdots, n)$ are constants.

Multiplying this vector x by A repeatedly, we have

$$\left.\begin{aligned} Ax &= \sum_{i=1}^{n} c_i \lambda_i u_i , \\ A^2x &= \sum_{i=1}^{n} c_i \lambda_i^2 u_i , \\ &\cdots\cdots \\ A^m x &= \sum_{i=1}^{n} c_i \lambda_i^m u_i = \lambda_1^m \left(c_1 u_1 + \sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1}\right)^m u_i\right) . \end{aligned}\right\} \tag{24}$$

Therefore, if

$$|\lambda_1| > |\lambda_2| \geqq |\lambda_3| \geqq \cdots \geqq |\lambda_n|,$$

the right side of the last equation in Eq. (24) approaches $c_1\lambda_1^m u_1$ when m approaches infinity, and the smaller the value of $|\lambda_2/\lambda_1|$ is, the faster its convergence will be. If

$$\lambda_1 = \lambda_2 \quad \text{and} \quad |\lambda_2| > |\lambda_3| \geqq \cdots \geqq |\lambda_n|,$$

the right side of last equation in Eq. (24) approaches $\lambda_1^m (c_1u_1 + c_2u_2)$ when m approaches infinity, because

$$A^m x = \lambda_1^m (c_1u_1 + c_2u_2 + \sum_{i=3}^{n} c_i (\frac{\lambda_i}{\lambda_1})^m u_i). \tag{25}$$

From the matters described aboved, it can be seen that $A^m x$ converges to the eigenvector, which corresponds to the maximum eigenvalue (the first eigenvalue λ_1), in all cases when m approaches infinity.

Next, from the first eigenvalue λ_1, the corresponding left eigenvector u_1 and right one u_1', we make a new matrix B as

$$B = A - \lambda_1 \frac{u_1 u_1'}{(u_1, u_1')} \tag{26}$$

where (u_1, u_1') is the inner product. The second eigenvalue λ_2 and the corresponding eigenvector u_2 can be obtained by applying Power Method to the matrix B. Furthermore, the other eigenvalues and eigenvectors can be obtained in turn by using the same way. However, it is clear that the error is stored and the accuracy decreases according to decrease of the eigenvalues.

Fig. 4 shows the flow chart for obtaining the eigenvalues by means of Power Method. In this figure, ϵ is the standard value which determines the completion of convergence to the eigenvalues and the eigenvectors.

In step 1. Eq. (11) is changed as

$$AX = \lambda X, \tag{27}$$

and Power Method is applied to this equation.

From the property of the correlation coefficient, n eigenvalues exist in the range of 0 to 1. Let $\lambda_1, \lambda_2, \cdots, \lambda_n$ be their eigenvalues under the condition $1 \geqq \lambda_1 \geqq \lambda_2 \geqq \cdots \geqq \lambda_n \geqq 0$. Since the speed of convergence to each eigenvalue and eigenvector depends on λ_2/λ_1 for $\lambda_1 \neq \lambda_2$, if the value of $|\lambda_1 - \lambda_2|$ is less than the constant ϵ, it is very difficult to converge because the number of iterative calculations considerably increases. However, it can be seen that the value of $|\lambda_1 - \lambda_2|$ is K/n at least, where K is a constant. (See Appendix.) From this fact, we get the following lemma:

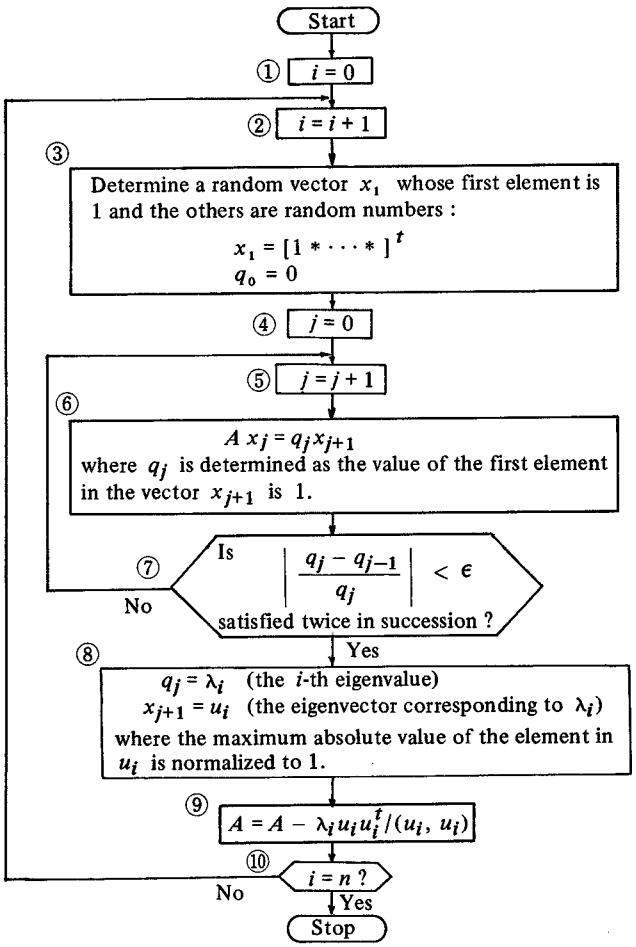

Fig. 4. The flow chart for obtaining the eigenvalues of an $n \times n$ symmetric matrix A.

[Lemma 1] When a graph of n vertices is represented in the graphic space, the number of times of the iterative calculations, m, required to obtain the eigenvalues and the eigenvectors with an accuracy $\epsilon = K/n$ and the number of figures, $P = K_0 + \log n$, where K_0 is a constant, by means of Power Method, is given by $m = O(n \log n)$.

(Proof) The condition for determining the convergence to the eigenvalue and the eigenvector is

$$(\lambda_2 / \lambda_1)^m < \epsilon = K/n .$$

Since the speed of convergence decreases according as the value of $|\lambda_1 - \lambda_2|$ decreases, in the case where $\lambda_1, \lambda_2, \cdots, \lambda_n$ exist with the same intervals of $1/n$ between 0 and 1, and the maximum eigenvalue λ_1 is 1, we get

$$\lambda_2/\lambda_1 = 1 - 1/n .$$

Then,

$$\lim_{n\to\infty} (1 - 1/n)^{n\ln n} = \lim_{n\to\infty} (1/e)^{\ln n} = 1/n .$$

This equation leads to

$$m = n\ln n = K n\log n = O(n\log n) .$$

Thus, the required solutions is obtained by carrying out the iterative calculations of $O(n\log n)$ times. QED.

In the block⑥in Fig. 4, we multiply the vector X by the $n \times n$ matrix A. Thus, we get the following lemma:

[Lemma 2] When a graph of n vertices is represented with an accuracy $\epsilon = K/n$ in the graphic space by Power Method, the processing time required to multiply once X by A under the number of figures, $P = K_0 + \log n$, is not more than $O(n^2 \log n)$.

From Lemmas 1 and 2, we obtain the following theorem:

[Theorem 3] The processing time to represent with an accuracy $\epsilon = K/n$ the graph of n vertices in the n dimensional graphic space is $O(n^4 (\log n)^2)$ at most.

In Theorem 3, if a given graph is not regular or has no any regular subgraph, that graph can be represented in the r dimensional graphic space, where r $(< n)$ is the constant which does not depend on n. Therefore, the processing time in this case is $O(n^3(\log n)^2)$ at most.

Since it is clear that the processing time in steps 2, 3 and 4 is less than $O(n^3)$, this algorithm for graph isomorphism can be done generally with the processing time of $O(n^4(\log n)^2)$ at most in total, and if the graph is not regular, it becomes $O(n^3(\log n)^2)$.

As an example, we now determine the isomorphism of two nondirected graphs G_a and G_b as shown in Fig. 5. These are the graphs made by removing two edges from a nondirected regular graph of degree 4 with 50 vertices and 100 edges, and the weights of their edges are all 1. If an accuracy ϵ takes the value 0.005, that is, $P = 4$ and $K_0 = 2 + \log 2$, all vertices of each graph can be represented separately in the 2 dimensional graphic space. Thus, the determination of isomorphism of these two graphs finished in 66.24 seconds. In Fig. 5, the corresponding vertices between G_a and G_b, which were determined by carrying out the algorithm, are shown by the same number associated with them.

5. Conclusions

In order to get an effective algorithm for graph isomorphism, an approach of representing the graphs in the r dimensional Euclidean space called graphic space was

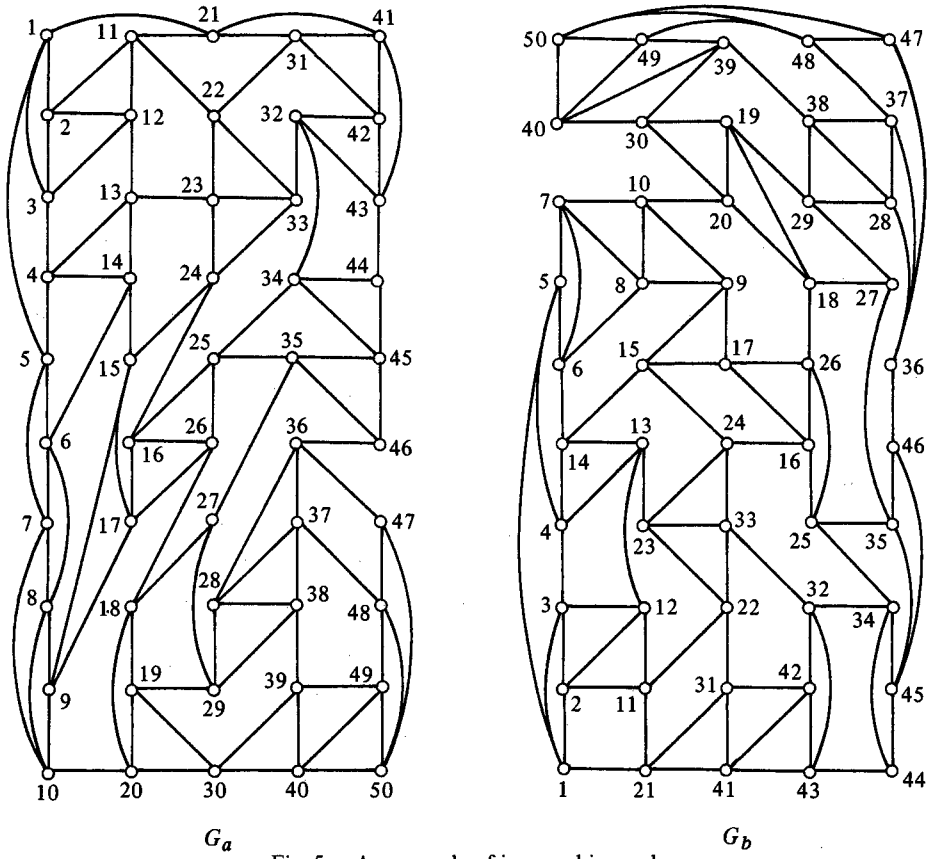

Fig. 5. An example of isomorphic graphs.

used, and the vertices of the given graphs were expressed as points in this space. Such a method for representing the graphs is also useful to survey the property of the graphs, for example, the center vertex, eccentricity and the like.

The distance among the vertices in the space, which depends on the strength of combination among them, gives a good contribution in determining the graph isomorphism. In other words, the distance between the graphs defined from that among the vertices in the graphic space is an important value for this purpose. The aigorithm described in this paper is based on the distance between the graphs in the graphic space, and if and only if the distance between two graphs in the space equals to zero, it is determined that the two graphs are isomorphic. Furthermore, the processing time of this algorithm was given as a function of the number of vertices, n, and it was also shown that the processing time was generally proportional to $n^4(\log n)^2$. Therefore, it is possible to use this algorithm for the graphs of realistic size. However, the processing time increases rapidly with increasing of n, so a subject for further discussion is to find the algorithm whose processing time is less than that of the algorithm described in this paper.

References

1) S. H. Unger, Comm. A. C. M., **7**, 26 (1964).
2) D. G. Corneil and C. C. Gotlieb, J. Assoc. Comp. Machinery, **17**, 51 (1970).
3) A. Mowshowitz, J. Combinational Theory, **12(B)**, 177 (1972).
4) M. Doob, Linear Albebra and its Applications, **3**, 461 (1970).
5) J. Turner, SIAM. J. Appl. Math., **16**, 520 (1968).
6) T. Kasai, K. Fukunaga and Y. Kitano, Lecture G7–18 in Annual Meeting of Kansai Branch of I.E.E. of Japan (Nov. 14, 1976).
7) K. Fukunaga, Y. Kitano and T. Kasai, Trans. IECE. Japan, **60-A**, 1093 (1977).
8) C. Hayashi, J. Inst. Statis. Math., **4**, 19 (1956).

Appendix

We now examine how far the value of $|\lambda_1 - \lambda_2|$ becomes small. The eigenvalues, $\lambda_1, \lambda_2, \cdots, \lambda_n$ $(1 \geqq \lambda_1 > \lambda_2 \geqq \lambda_3 \geqq \cdots \geqq \lambda_n > 0)$, of the eigen equation,

$$AX = \lambda CX , \tag{A-1}$$

are obtained from

$$|A - \lambda C| = 0 . \tag{A-2}$$

Since Eq. (A-2) can be changed into

$$x^k (a'_{n-k} x^{n-k} + a'_{n-k-1} x^{n-k-1} + \cdots + a_1' x + a_0') = 0 , \tag{A-3}$$

the eigenvalues λ_1 and λ_2 are obtained as the solutions of

$$\sum_{h=0}^{n-k} a'_h x^h = 0 , \tag{A-4}$$

where $a'_{n-k} \neq 0$.

Now let the elements of the matrices A and C in Eq. (A-1), and the coefficients a'_h $(h = 0, 1, \cdots, n-k)$ in Eq. (A-3) be expressed with P figures. If Eq. (A-4) is divided by the coefficient, a'_{n-k}, of the highest order, it becomes

$$x^{n-k} + \sum_{h=0}^{n-k-1} a_h x^h = 0 , \tag{A-5}$$

where $a_h = a'_h / a'_{n-k}$ $(h = 0, 1, \cdots, n-k-1)$. It is clear that the coefficients in the above equation take the following values:

$$a_h = 0 \quad \text{or} \quad |a_h| > 1/10^P \quad (h = 0, 1, \cdots, n-k-1). \tag{A-6}$$

Suppose $n - k = 2$. Then, from Eq. (A-5)

$$|\lambda_1 - \lambda_2| = \sqrt{a_1^2 - 4a_0} . \tag{A-7}$$

Since $\lambda_1 \neq \lambda_2$,

$$\text{MIN}\,|\lambda_1 - \lambda_2| = |a_1| > 1/10^P .$$

If $P = K_0 + \log n$,

$$\text{MIN} \mid \lambda_1 - \lambda_2 \mid \; > \; 1/10^{K_0 + \log n} \; = \; K/n \, .$$

For $n - k \geqq 3$, Eq. (A-5) is solved into factors as

$$\prod_{h=1}^{n-k} (x - b_h) \; = \; 0 \, . \tag{A-8}$$

From the relationship of the solutions and the coefficients, we have

$$0 \; < \; b_h \; \leqq \; 1 \qquad (h \; = \; 1, 2, \cdots, n-k) \tag{A-9}$$

and

$$\prod_{h=1}^{n-k} b_h \; = \; a_0 \; > \; 1/10^P \, . \tag{A-10}$$

It is necessary that two eigenvalues λ_1 and λ_2 satisfy the conditions of Eqs. (A-9) and (A-10), and that one of them approaches most closely to the other. Thus, if $b_3 = b_4 = \cdots = b_{n-k} = 1$ for the first eigenvalue λ_1, the result obtained from Eq. (A-8) is the same as that in the case where $n - k = 2$.

From the above considerations, we may conclude that the difference between two eigenvalues, $\mid \lambda_1 - \lambda_2 \mid$, of Eq. (A-1) is K/n at least.