An Eigendecomposition Approach to Weighted Graph Matching Problems

SHINJI UMEYAMA

Abstract—This paper discusses an approximate solution to the weighted graph matching problem (WGMP) for both undirected and directed graphs. The WGMP is the problem of finding the optimum matching between two weighted graphs, which are graphs with weights at each arc. The proposed method employs an analytic, instead of a combinatorial or iterative, approach to the optimum matching problem of such graphs. By using the eigendecompositions of the adjacency matrices (in the case of the undirected graph matching problem) or some Hermitian matrices derived from the adjacency matrices (in the case of the directed graph matching problem), a matching close to the optimum one can be found efficiently when the graphs are sufficiently close to each other. Simulation experiments are also given to evaluate the performance of the proposed method.

Index Terms—Eigendecomposition, inexact matching, structural description, structural pattern recognition, weighted graph matching.

I. Introduction

STRUCTURAL description, i.e., the description of a certain object in terms of its parts, their properties, and their mutual relations, is one of the most general methods for representing the real world. Properties and relations may be qualitative (symbolic values) or quantitative (numerical values). For example, in the interpretation of visual images, a "man" can be described in terms of his parts (head, body, arms, etc.), their properties (shape, color, etc.), and their relations (head is above body, etc.). Another simple example is a point pattern in 2-D or 3-D space. A point pattern can be represented by using a structural description if we employ distances between points as their mutual relations and point labels (if they exist) as their properties.

There are many problems in handling structural descriptions, for example, construction of a description from the given data, classification of the given descriptions, etc. Above all, one of the most difficult but interesting problems is optimum matching between structural descriptions, i.e., the problem of finding the correspondence between their parts in order to make the corresponding properties and relations as consistent as possible [1]. This problem is fundamental in various fields, especially in the fields of image analysis and computer vision. For example, matching feature point descriptions, derived from a sensing system, with known models stored in a database is considered to be an essential part of a computer vision system [2].

A great deal of material dealing with this problem has been published. You [3] employed a combinatorial (tree search) approach and applied the branch-and-bound algorithm to this problem. The algorithm always gives the true optimum matching; however, it may be impractical in analyzing highly complex structures because of its inherent combinatorial property. Tsai and Fu [1] also described a tree search technique for finding isomorphisms between graphs which bear both symbolic and numerical labels. Kitchen [4], [5] solved this problem in both qualitative and quantitative cases by using a relaxation method. The method is tolerant to noise, and it can take advantage of hardware parallelism. However, the relaxation method is inherently a local optimization method, and it is sometimes difficult to determine a suitable updating rule (the local compatibility function).

A structural description can be represented by an attributed or labeled (hyper-) graph. A weighted graph, i.e., a graph with weights at each arc, is a simple example of a quantitative structural description. In this paper, we treat weighted graphs with the same number of nodes and employ the analytic, instead of combinatorial or iterative, approach to the optimum matching problem of such graphs. By using the eigendecompositions of the adjacency matrices (in the case of the undirected graph matching problem) or some Hermitian matrices derived from the adjacency matrices (in the case of the directed graph matching problem), a matching close to the optimum one can be found efficiently when the graphs are sufficiently close to each other.

An ordinary graph in graph theory can be considered to be a weighted graph with a binary weighting function. Thus, the weighted graph matching problem (WGMP) includes the graph isomorphism problem, which is proved neither to be NP-complete nor to have an efficient algorithm [6]. It is possible to apply the same method proposed here to the graph isomorphism problem.

II. Weighted Graph Matching Problem (WGMP)

A. Preliminary

A weighted graph G is an ordered pair (V, w) where V is a set of nodes of the graph and w is a weighting function which gives a real nonnegative value $w(v_i, v_j)$ to each

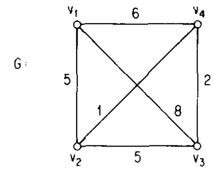
Manuscript received March 31, 1986; revised January 15, 1988.
The author is with the Mathematical Engineering Section, Computer Science Division, Electrotechnical Laboratory, 1-1-4 Umezono, Tsukuba-shi, Ibaraki 305, Japan.
IEEE Log Number 8823194.

pair of nodes $(v_i v_j)$ (arc of the graph), $v_i \in V$, $v_j \in V$, and $v_i \neq v_j$. We assume $|V| = n$. The nodes of a graph are supposed to be labeled by names such as $v_1, v_2, \cdots, v_n$ to be distinguished from one another. The weighted graph is called undirected when its weighting function is symmetric, i.e., $w(v_i, v_j) = w(v_j, v_i)$ for all $v_i, v_j, v_i \neq v_j$. Otherwise, it is called directed. Obviously, an ordinary graph is a weighted graph with a binary weighting function. When a weighting function takes a value between 0 and 1, such a graph is sometimes called a fuzzy graph [7]. However, here we do not place such a restriction on the range of the weighting function; i.e., it can take any nonnegative real value.

The adjacency matrix of a weighted graph $G = (V, w)$ is an $n \times n$ matrix A_G defined as follows:

$$A_G = [a_{ij}]\begin{cases} a_{ij} = w(v_i, v_j) & i \neq j \\ a_{ii} = 0. \end{cases} \tag{1}$$

When G is undirected, A_G becomes a symmetric matrix. Fig. 1 shows an example of a weighted undirected graph G with four nodes and its adjacency matrix A_G.



$$A_G = \begin{bmatrix} 0.0 & 5.0 & 8.0 & 6.0 \\ 5.0 & 0.0 & 5.0 & 1.0 \\ 8.0 & 5.0 & 0.0 & 2.0 \\ 6.0 & 1.0 & 2.0 & 0.0 \end{bmatrix}$$

Fig. 1. A weighted undirected graph G and its adjacency matrix A_G.

B. Definitions of the WGMP

Let $G = (V_1, w_1)$, $H = (V_2, w_2)$ be weighted graphs with n nodes. The WGMP is the problem of finding a one-to-one correspondence Φ between $V_1 = \{v_1, v_2, \cdots, v_n\}$ and $V_2 = \{v_1', v_2', \cdots, v_n'\}$ which minimizes a "difference" between G and H. In this paper, we use the following criterion for a measure of difference:

$$J(\Phi) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(w_1(v_i, v_j) - w_2(\Phi(v_i), \Phi(v_j)) \right)^2. \tag{2}$$

If G and H are weighted graphs and A_G and A_H are their adjacency matrices, respectively, $J(\Phi)$ can be reformulated as follows by using a permutation matrix P:

$$J(P) = \left\| P A_G P^T - A_H \right\|^2 \tag{3}$$

where the permutation matrix P represents the node correspondence Φ and $\|\cdot\|$ is the Euclidean norm ($\|A\| = (\Sigma_{i=1}^{n} \Sigma_{j=1}^{n} |a_{ij}|^2)^{1/2}$). Thus, the WGMP is reduced to the problem of finding the permutation matrix P which minimizes $J(P)$.

Two ordinary graphs are called isomorphic if there exists a one-to-one correspondence between their node sets, which preserves adjacency. Similarly, we call two weighted graphs G and H isomorphic if there exists a one-to-one correspondence Φ which makes $J(\Phi)$ equal to 0. Thus, in this case we have

$$P A_G P^T = A_H. \tag{4}$$

Because ordinary graphs are weighted graphs with a binary weighting function, the WGMP includes the graph isomorphism problem.

In general, it is not easy to find the exact solution of the WGMP since the WGMP is a purely combinatorial problem. Thus, we need to develop an efficient method which gives a "nearly" optimum solution, i.e., a permutation matrix P' whose criterion value $J(P')$ is very close to the optimum value. Of course, since we cannot know the optimum criterion value in advance, what we can do is to give an algorithm which determines a permutation matrix of a small criterion value. The purpose of this paper is to propose a fast matching algorithm giving such permutation matrices.

Note that the WGMP of graphs with weights at each node as well as at each arc can be formulated and solved in a similar way by defining $a_{ii} = w(i, i)$ in (1) where $w(i, i)$ is a weight at node i.

III. Weighted Undirected Graph Matching Algorithm

In this section, we show an algorithm that gives a nearly optimum solution to the weighted undirected graph matching problem.

Let G and H be weighted undirected graphs and A_G and A_H be their adjacency matrices, respectively. The optimum matching between G and H is a permutation matrix P which minimizes $J(P)$ in (3). It is, in general, difficult to find this matrix P directly. However, if we extend the domain of J to the set of orthogonal matrices, the orthogonal matrices (Q) which minimize $J(Q)$ can be obtained in closed forms by using the eigendecomposition of the adjacency matrices A_G and A_H. This extension of the domain is very natural because a permutation matrix is a kind of orthogonal matrix. A nearly optimum permutation matrix is determined by using these orthogonal matrices as clues.

Here, we assume that both adjacency matrices A_G and A_H have n distinct eigenvalues, respectively. This is not a strong assumption for real data. Besides that, even if they have multiple roots, this can be overcome by perturbing them. Small perturbations will not effect the result of matching.

A. Determination of Nearly Optimum Permutation Matrix

First, we give Theorem 1 [8].

Theorem 1: If A and B are Hermitian matrices ($A = A^*$ and $B = B^*$) with eigenvalues $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$

and $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$, respectively, then

$$\|A - B\|^2 \geq \sum_{i=1}^{n} (\alpha_i - \beta_i)^2. \quad (5)$$

From Theorem 1, we have the following theorem.

Theorem 2: Let A and B be $n \times n$ (real) symmetric matrices with n distinct eigenvalues $\alpha_1 > \alpha_2 > \cdots > \alpha_n$ and $\beta_1 > \beta_2 > \cdots > \beta_n$, respectively, and their eigendecompositions be given by

$$A = U_A \Lambda_A U_A^T, \quad (6)$$

$$B = U_B \Lambda_B U_B^T \quad (7)$$

where U_A and U_B are orthogonal matrices and $\Lambda_A = \mathrm{diag}\,(\alpha_i)$ and $\Lambda_B = \mathrm{diag}\,(\beta_i)$. Then, the minimum of $\|QAQ^T - B\|^2$, where Q ranges over the set of all orthogonal matrices, is attained for the following Q's:

$$Q = U_B S U_A^T,$$

$$S \in \mathbb{S}_1 = \left\{ \mathrm{diag}\,(s_1, s_2, \cdots, s_n) \,|\, s_i = 1 \text{ or } -1 \right\}. \quad (8)$$

The minimum value achieved is $\Sigma_{i=1}^{n} (\alpha_i - \beta_i)^2$.

Proof: From Theorem 1, (9) holds for any orthogonal matrix R since eigenvalues of RAR^T are the same as those of A.

$$\|RAR^T - B\|^2 \geq \sum_{i=1}^{n} (\alpha_i - \beta_i)^2. \quad (9)$$

On the other hand, if we use Q in (8), we have

$$\begin{aligned} \|QAQ^T - B\|^2 &= \| U_B S U_A^T U_A \Lambda_A U_A^T U_A S U_B^T - U_B \Lambda_B U_B^T \|^2 \\ &= \| U_B (S \Lambda_A S - \Lambda_B) U_B^T \|^2 \\ &= \| S \Lambda_A S - \Lambda_B \|^2 \\ &= \| \Lambda_A - \Lambda_B \|^2 \\ &= \sum_{i=1}^{n} (\alpha_i - \beta_i)^2 \end{aligned} \quad (10)$$

where we used the equations that $\|UX\| = \|XU^T\| = \|X\|$ for any orthogonal matrix U and $S\Lambda_A S = S^2 \Lambda_A = \Lambda_A$ since S and Λ_A are both diagonal matrices and $S^2 = I$. Thus, the minimum of $\|QAQ^T - B\|^2$ is attained for Q's given in (8). □

The orthogonal matrices which minimize $J(Q)$ in (3) are given from Theorem 2 by

$$Q = U_H S U_G^T, \qquad S \in \mathbb{S}_1, \quad (11)$$

assuming eigendecompositions of A_G and A_H as

$$A_G = U_G \Lambda_G U_G^T, \quad (12)$$

$$A_H = U_H \Lambda_H U_H^T. \quad (13)$$

Now we assume that G and H are isomorphic. In this case, we have from (4), (12), and (13) that

$$P U_G \Lambda_G U_G^T P^T = U_H \Lambda_H U_H^T. \quad (14)$$

Thus,

$$P U_G = U_H S, \qquad S \in \mathbb{S}_1 \quad (15)$$

since the eigenvectors of a matrix are uniquely determined, except for their positive and negative directions when all eigenvalues are distinct (when x is an eigenvector of a matrix A, $-x$ is also an eigenvector of A). Then,

$$P = U_H S U_G^T. \quad (16)$$

This means that there exists some $S \in \mathbb{S}_1$ which exactly makes Q a permutation matrix when G and H are isomorphic. We write this diagonal matrix as $\hat{S}$, and $\hat{P} = U_H \hat{S} U_G^T$. Let $U_H = [h_{ij}]$, $U_G = [g_{ij}]$, $\hat{S} = \mathrm{diag}\,(s_i)$, and

$$\hat{P} = \begin{array}{c} \\ i \\ \\ \end{array} \begin{array}{c} \pi(i) \\ \left(\begin{array}{cc} & \vdots \\ \cdots & 1 \end{array} \right) \end{array}. \quad (17)$$

Then we have

$$\mathrm{tr}\,(\hat{P}^T U_H \hat{S} U_G^T) = \sum_{i=1}^{n} \sum_{k=1}^{n} s_k h_{ik} g_{\pi(i),k}. \quad (18)$$

Obviously, the following holds:

$$\begin{aligned} \sum_{k=1}^{n} s_k h_{ik} g_{\pi(i),k} &\leq \left| \sum_{k=1}^{n} s_k h_{ik} g_{\pi(i),k} \right| \\ &\leq \sum_{k=1}^{n} \left| s_k h_{ik} g_{\pi(i),k} \right| \\ &= \sum_{k=1}^{n} |h_{ik}| |g_{\pi(i),k}|. \end{aligned} \quad (19)$$

Thus,

$$\begin{aligned} \mathrm{tr}\,(\hat{P}^T U_H \hat{S} U_G^T) &\leq \sum_{i=1}^{n} \sum_{k=1}^{n} |h_{ik}| |g_{\pi(i),k}| \\ &= \mathrm{tr}\,(\hat{P}^T \overline{U}_H \overline{U}_G^T) \end{aligned} \quad (20)$$

where $\overline{U}_H$ and $\overline{U}_G$ are matrices which have as each element the absolute value of each element of U_H and U_G, respectively. Since the length of each row vector of $\overline{U}_G$ and $\overline{U}_H$ is equal to 1 and the values of its elements are nonnegative, the following holds for each ij-element x_{ij} (scalar product of the ith row of $\overline{U}_H$ and the jth row of $\overline{U}_G$) of $\overline{U}_H \overline{U}_G^T$:

$$0 \leq x_{ij} \leq 1. \quad (21)$$

Thus, we have

$$\mathrm{tr}\,(P^T \overline{U}_H \overline{U}_G^T) \leq n \quad (22)$$

for any permutation matrix P. On the other hand, the following holds, obviously:

$$\begin{aligned} \mathrm{tr}\,(\hat{P}^T U_H \hat{S} U_G) &= \mathrm{tr}\,(\hat{P}^T \hat{P}) \\ &= n. \end{aligned} \quad (23)$$

Thus, from (20), (22), and (23), we have

$$\mathrm{tr}\,(\hat{P}^T \overline{U}_H \overline{U}_G^T) \le n. \tag{24}$$

This means that $\hat{P}$ maximizes tr $(P^T \overline{U}_H \overline{U}_G^T)$ since tr $(P^T \overline{U}_H \overline{U}_G^T) \le n$ for any permutation matrix P. Therefore, when G and H are isomorphic, the optimum permutation matrix can be obtained as a permutation matrix P which maximizes tr $(P^T \overline{U}_H \overline{U}_G^T)$. This problem is an instance of the assignment (bipartite maximum weighted matching) problem, and can be solved by the Hungarian method, running in $O(n^3)$ time [9].

Next, we consider the case when G and H are not isomorphic. From the argument above, even if G and H are not isomorphic, it can be expected that the permutation matrix P which maximizes tr $(P^T \overline{U}_H \overline{U}_G^T)$ will be very close to the optimum permutation matrix when G and H are nearly isomorphic. Thus, this permutation matrix P can be used as a candidate for the optimum one. This may be valid only when G and H are nearly isomorphic. However, even if G and H are not sufficiently close to each other, the global correspondence between G and H should be reflected in P. Thus, this permutation matrix P gives a good initial correspondence when we improve it, for example, by hill-climbing or relaxation methods.

B. Example for Weighted Undirected Graph Matching

We give an artificial example of our method to clarify the algorithm. Fig. 2 shows the weighted undirected graphs G and H with four nodes. The adjacency matrices of G and H are A_G and A_H given in (25) and (26):

$$A_G = \begin{pmatrix} 0.0 & 5.0 & 8.0 & 6.0 \\ 5.0 & 0.0 & 5.0 & 1.0 \\ 8.0 & 5.0 & 0.0 & 2.0 \\ 6.0 & 1.0 & 2.0 & 0.0 \end{pmatrix}, \tag{25}$$

$$A_H = \begin{pmatrix} 0.0 & 1.0 & 8.0 & 4.0 \\ 1.0 & 0.0 & 5.0 & 2.0 \\ 8.0 & 5.0 & 0.0 & 5.0 \\ 4.0 & 2.0 & 5.0 & 0.0 \end{pmatrix}. \tag{26}$$

Eigendecompositions of A_G and A_H are given as follows:

$$A_G = U_G \Lambda_G U_G^T \tag{27}$$

$$\Lambda_G = \mathrm{diag}\,(14.25, -0.28, -4.83, -9.14),$$

$$U_G = \begin{pmatrix} 0.614 & 0.141 & -0.182 & -0.755 \\ 0.434 & -0.528 & 0.726 & 0.079 \\ 0.548 & -0.270 & -0.582 & 0.536 \\ 0.366 & 0.793 & 0.317 & 0.369 \end{pmatrix},$$

$$A_H = U_H \Lambda_H U_H^T \tag{28}$$

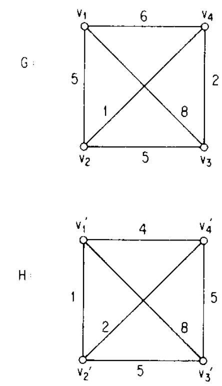


Fig. 2. An example of the weighted undirected graph matching problem.

$$\Lambda_H = \mathrm{diag}\,(13.26, -0.77, -3.43, -9.05),$$

$$U_H = \begin{pmatrix} 0.538 & -0.436 & -0.425 & -0.583 \\ 0.344 & 0.880 & -0.018 & -0.327 \\ 0.624 & 0.026 & -0.250 & 0.740 \\ 0.450 & -0.187 & 0.870 & -0.079 \end{pmatrix}.$$

From this, we have

$$\overline{U}_G = \begin{pmatrix} 0.614 & 0.141 & 0.182 & 0.755 \\ 0.434 & 0.528 & 0.726 & 0.079 \\ 0.548 & 0.270 & 0.582 & 0.536 \\ 0.366 & 0.793 & 0.317 & 0.369 \end{pmatrix}, \tag{29}$$

$$\overline{U}_H = \begin{pmatrix} 0.538 & 0.436 & 0.425 & 0.583 \\ 0.344 & 0.880 & 0.018 & 0.327 \\ 0.624 & 0.026 & 0.250 & 0.740 \\ 0.450 & 0.187 & 0.870 & 0.079 \end{pmatrix}. \tag{30}$$

Thus,

$$\overline{U}_H \overline{U}_G^T = \begin{pmatrix} 0.909 & 0.818 & 0.973 & 0.893 \\ 0.585 & 0.653 & 0.612 & 0.950 \\ 0.991 & 0.524 & 0.892 & 0.601 \\ 0.520 & 0.931 & 0.846 & 0.618 \end{pmatrix}. \tag{31}$$

The permutation matrix P obtained by applying the Hungarian method to (31) is

$$P = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{32}$$

Hence, v_1 in G corresponds to v_3' in H, v_2 corresponds to v_4', v_3 corresponds to v_1', and v_4 corresponds to v_2'. This is the optimum matching.

IV. Weighted Directed Graph Matching Algorithm

We give the algorithm for the weighted directed graph matching problem in this section. Let G and H be weighted directed graphs and A_G and A_H be their adjacency matrices, respectively.

A nearly optimum solution for the weighted undirected graph matching problem was given in the previous section by using the eigendecompositions of adjacency matrices. This method cannot be applied directly to the case of the directed graphs since adjacency matrices are no longer symmetric in this case. However, we can also give a nearly optimum solution to the weighted directed graph matching problem in a way similar to the case of the undirected graphs if we use some Hermitian matrices derived from the adjacency matrices.

A. Determination of Nearly Optimum Permutation Matrix

Any matrix can be decomposed uniquely into a sum of a symmetric and a skew-symmetric matrix; thus, adjacency matrices A_G and A_H can be decomposed as follows:

$$A_{GS} = (A_G + A_G^T)/2, \tag{33}$$

$$A_{GN} = (A_G - A_G^T)/2, \tag{34}$$

$$A_{HS} = (A_H + A_H^T)/2, \tag{35}$$

$$A_{HN} = (A_H - A_H^T)/2 \tag{36}$$

where A_{GS} and A_{HS} are symmetric components of A_G and A_H, and A_{GN} and A_{HN} are skew-symmetric components of A_G and A_H, respectively. Complex matrices E_G and E_H are defined from these matrices as follows:

$$E_G = A_{GS} + iA_{GN}, \tag{37}$$

$$E_H = A_{HS} + iA_{HN}. \tag{38}$$

Obviously, E_G and E_H are Hermitian matrices.

$$\begin{aligned} E_G^* &= A_{GS}^T - iA_{GN}^T \\ &= A_{GS} + iA_{GN} \\ &= E_G, \end{aligned} \tag{39}$$

$$\begin{aligned} E_H^* &= A_{HS}^T - iA_{HN}^T \\ &= A_{HS} + iA_{HN} \\ &= E_H. \end{aligned} \tag{40}$$

Here, we introduce a new criterion J' using E_G and E_H.

$$J'(P) = \|PE_GP^T - E_H\|^2. \tag{41}$$

Substituting (37) and (38) into (41), we have

$$\begin{aligned} J'(P) &= \left\|P(A_{GS} + iA_{GN})P^T - (A_{HS} + iA_{HN})\right\|^2 \\ &= \left\|(PA_{GS}P^T - A_{HS}) + i(PA_{GN}P^T - A_{HN})\right\|^2 \\ &= \|PA_{GS}P^T - A_{HS}\|^2 + \|PA_{GN}P^T - A_{HN}\|^2. \end{aligned} \tag{42}$$

Therefore, a permutation matrix P which minimizes $J'(P)$ is the permutation matrix which minimizes the differences of the symmetric and the skew-symmetric components of adjacency matrices at the same time.

This matrix P is not, in general, the optimum permutation matrix which minimizes $J(P)$ in (3). However, we adopt here $J'(P)$ as a criterion instead of $J(P)$ because of its simplicity. $J'(P)$ is usually equivalent to $J(P)$ when G and H are nearly isomorphic, and they are exactly equivalent when G and H are isomorphic or if G and H are not isomorphic, but undirected.

From Theorem 1, we have the following theorem.

Theorem 3: Let C and D be $n \times n$ Hermitian matrices with n distinct eigenvalues $\gamma_1 > \gamma_2 > \cdots > \gamma_n$ and $\delta_1 > \delta_2 > \cdots > \delta_n$, respectively, and their eigendecompositions be given by

$$C = W_C \Gamma_C W_C^*, \tag{43}$$

$$D = W_D \Gamma_D W_D^* \tag{44}$$

where W_C and W_D are unitary matrices and $\Gamma_C = \text{diag}(\gamma_i)$ and $\Gamma_D = \text{diag}(\delta_i)$. Then, the minimum of $\|TCT^* - D\|^2$, where T ranges over the set of all unitary matrices, is attained for the following T's:

$$\begin{aligned} &T = W_D S W_C^*, \\ &S \in \mathcal{S}_2 = \{\text{diag}(s_1, s_2, \cdots, s_n) \mid \\ &\qquad s_i \text{ is any complex number satisfying} \\ &\qquad |s_i| = 1\}. \end{aligned} \tag{45}$$

The minimum value achieved is $\Sigma_{i=1}^n (\gamma_i - \delta_i)^2$.

The proof of this theorem can be obtained in a way similar to that of Theorem 2.

We assume here that both E_G and E_H have n distinct eigenvalues for the same reason as in the weighted undirected matching problem and give their eigendecompositions as follows:

$$E_G = W_G \Gamma_G W_G^*, \tag{46}$$

$$E_H = W_H \Gamma_H W_H^*. \tag{47}$$

Then, from Theorem 3 and an argument similar to that of the previous section (in this case, the domain of J' is extended to the set of unitary matrices), a nearly optimum permutation matrix can be given as the permutation matrix P which maximizes tr $(P^T \overline{W}_H \overline{W}_G^T)$. This is obtained by applying the Hungarian method to $\overline{W}_H \overline{W}_G^T$.

B. Example of Weighted Directed Graph Matching

We give again an artificial example of our method to clarify the algorithm. Fig. 3 shows the weighted directed graphs G and H with four nodes. The adjacency matrices of G and H are A_G and A_H, given in (48) and (49):

$$A_G = \begin{pmatrix} 0.0 & 3.0 & 4.0 & 2.0 \\ 0.0 & 0.0 & 1.0 & 2.0 \\ 1.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{pmatrix}, \tag{48}$$

$$A_H = \begin{pmatrix} 0.0 & 4.0 & 2.0 & 4.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 2.0 & 0.0 & 2.0 \\ 0.0 & 1.0 & 2.0 & 0.0 \end{pmatrix}. \tag{49}$$

E_G and E_H are as follows:

$$E_G = \begin{pmatrix} 0.0 & 1.5 & 2.5 & 1.0 \\ 1.5 & 0.0 & 0.5 & 1.0 \\ 2.5 & 0.5 & 0.0 & 1.0 \\ 1.0 & 1.0 & 1.0 & 0.0 \end{pmatrix} + i \begin{pmatrix} 0.0 & 1.5 & 1.5 & 1.0 \\ -1.5 & 0.0 & 0.5 & 1.0 \\ -1.5 & -0.5 & 0.0 & 0.0 \\ -1.0 & -1.0 & 0.0 & 0.0 \end{pmatrix}, \tag{50}$$

$$E_H = \begin{pmatrix} 0.0 & 2.0 & 1.0 & 2.0 \\ 2.0 & 0.0 & 1.5 & 0.5 \\ 1.0 & 1.5 & 0.0 & 2.0 \\ 2.0 & 0.5 & 2.0 & 0.0 \end{pmatrix} + i \begin{pmatrix} 0.0 & 2.0 & 1.0 & 2.0 \\ -2.0 & 0.0 & -0.5 & -0.5 \\ -1.0 & 0.5 & 0.0 & 0.0 \\ -2.0 & 0.5 & 0.0 & 0.0 \end{pmatrix}. \tag{51}$$

Eigendecompositions of E_G and E_H are given as follows:

$$E_G = W_G \Gamma_G W_G^*,$$

$$\Gamma_G = \mathrm{diag}\,(4.76,\ 0.11,\ -1.34,\ -3.54),$$

$$W_G = \begin{pmatrix} 0.38 & 0.19 & 0.03 & 0.70 \\ 0.40 & -0.05 & -0.26 & -0.11 \\ 0.50 & 0.17 & -0.23 & -0.46 \\ 0.41 & -0.49 & 0.76 & -0.12 \end{pmatrix} + i \begin{pmatrix} 0.49 & 0.25 & -0.12 & -0.11 \\ 0.18 & -0.61 & -0.43 & 0.40 \\ 0.11 & 0.50 & 0.32 & 0.31 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{pmatrix}, \tag{52}$$

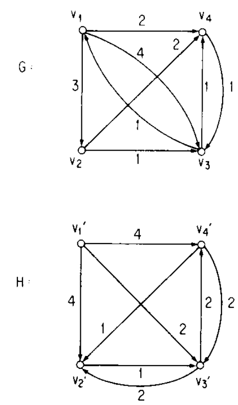


Fig. 3. An example of the weighted directed graph matching problem.

$$E_H = W_H \Gamma_H W_H^*,$$

$$\Gamma_H = \mathrm{diag}\,(5.69,\ -0.35,\ -1.39,\ -3.95),$$

$$W_H = \begin{pmatrix} 0.45 & -0.25 & 0.39 & -0.43 \\ 0.45 & -0.70 & -0.19 & 0.49 \\ 0.45 & 0.27 & -0.71 & -0.20 \\ 0.50 & 0.58 & 0.38 & 0.52 \end{pmatrix} + i \begin{pmatrix} 0.37 & 0.07 & 0.08 & -0.49 \\ -0.11 & 0.05 & -0.08 & 0.11 \\ 0.00 & -0.17 & 0.38 & -0.09 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{pmatrix}. \tag{53}$$

From this, we have

$$\overline{W}_G = \begin{pmatrix} 0.616 & 0.312 & 0.121 & 0.713 \\ 0.439 & 0.614 & 0.507 & 0.417 \\ 0.511 & 0.531 & 0.392 & 0.550 \\ 0.408 & 0.494 & 0.758 & 0.121 \end{pmatrix}, \tag{54}$$

$$\overline{W}_H = \begin{pmatrix} 0.583 & 0.264 & 0.403 & 0.654 \\ 0.463 & 0.702 & 0.205 & 0.501 \\ 0.447 & 0.315 & 0.808 & 0.221 \\ 0.495 & 0.582 & 0.379 & 0.522 \end{pmatrix}. \tag{55}$$

Thus,

$$\overline{W}_H \overline{W}_G^T = \begin{pmatrix} 0.957 & 0.895 & 0.956 & 0.753 \\ 0.886 & 0.947 & 0.966 & 0.751 \\ 0.629 & 0.891 & 0.834 & 0.977 \\ 0.905 & 0.984 & 0.998 & 0.840 \end{pmatrix}. \quad (56)$$

The permutation matrix P obtained by applying the Hungarian method to (56) is

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \quad (57)$$

This is the optimum matching.

V. Performance Evaluation of the Weighted Graph Matching Algorithm

Simulation experiments are presented here to illustrate the performance of the proposed weighted graph matching algorithm. Pairs of weighted graphs are generated stochastically and matched with each other by using several matching algorithms.

Weights of a graph G were produced by a pseudorandom number generator, which assigns a real number uniformly distributed in the range of 0–1.0 to each arc. A graph H was created by modifying the graph G, i.e., by adding uniformly distributed noise in the range of $-\epsilon$ to $+\epsilon$ to each weight of G and shuffling the order of nodes, or by producing a new H in the same way as G (we indicate this case by writing ϵ = "random").

Graph pairs of different sizes and different noise levels (ϵ) were produced as input to the algorithms. Sizes ranging from 5 to 10 and noise levels ranging from 0 to 0.2 or ϵ = random have been tested. For each parameter, 50 pairs of weighted (undirected and directed) graphs were generated and matched with each other by the following four algorithms:

a) the proposed weighted graph matching algorithm;

b) a tree search algorithm (branch-and-bound method) proposed by You [3], which was used to obtain real optimum matchings;

c) the hill-climbing method, i.e., improving a randomly generated initial correspondence by exchanging node correspondences locally so that the value of the criterion [eq. (3)] decreases (see Fig. 4); searching stops if the criterion cannot be improved any more; and

d) the hill-climbing method with an initial correspondence obtained by the proposed weighted graph matching algorithm.

The tree search algorithm was not tested on graphs with sizes greater than 7 because of the CPU time limit. The sample means and standard deviations of the criterion values and the total CPU time elapsed during the matching process were recorded for each parameter. $J(P)$ in (3) was used to compute the criterion value for the cases of both undirected and directed graph matchings, although $J'(P)$ in (42) is used in the directed graph matching algorithm. The number of real optimum matchings obtained by each method was also recorded. The experiments were done on a VAX-11/780.

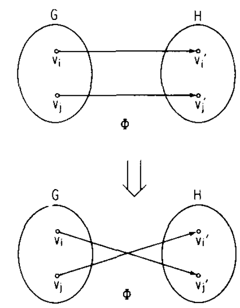


Fig. 4. Local exchange of node correspondences.

Tables I and II show the sample means and standard deviations of the criterion values and the numbers of real optimum matchings (k) obtained by each method for several noise levels (ϵ = 0.0, 0.05, 0.10, 0.15, 0.20, and random) where the sizes of the generated graphs are n = 7 and 10. Table I is for the case of undirected graphs, and Table II is for the case of directed graphs. The number of real optimum matchings obtained is not shown in the table when the size of the graphs is $n = 10$ since method b) (the tree search algorithm) was not tested, and the real optimum matchings are not known in this case. The last column of each table shows the expected values of the criterion. Supposing that the optimum matching is not affected by adding noise, the expected value $E(n, \epsilon)$ of the criterion is given as follows:

$$E(n, \epsilon) = \epsilon^2 n(n-1)/3 \quad (58)$$

since the variance of the noise uniformly distributed in the range of $-\epsilon$ to $+\epsilon$ is equal to $\epsilon^2/3$.

Tables III and IV give the sample means and standard deviations of the total CPU time in milliseconds, elapsed during the matching process for several graph sizes (n = 5, 6, 7, 8, 9, and 10) when the noise levels were $\epsilon = 0.1$ and = 0.2. Table III is for the case of undirected graphs, and Table IV is for the case of directed graphs.

From these tables, the following is obvious. A combinatorial method [method b)] seems impractical for large problems (matchings between large graphs) because of its time. The hill-climbing method [method c)] may be applicable to large problems since it is not too slow. However, this method often gives local optima because it is a kind of local optimization method. The proposed matching method [method a)] is the fastest method among the methods in these experiments and almost always gives the real optimum matching when a pair of graphs are nearly isomorphic ($\epsilon \leq 0.1$). However, if a pair of graphs are

TABLE I
PERFORMANCE EVALUATION OF THE WEIGHTED UNDIRECTED GRAPH MATCHING ALGORITHMS (50 TRIALS PER DATUM)[a]

n = 7

ε	Method (a) Mean(S.D.) k	Method (b) Mean(S.D.) k	Method (c) Mean(S.D.) k	Method (d) Mean(S.D.) k	$E(n, \varepsilon)$
0.00	0.00(0.00) 50	0.00(0.00) 50	1.32(1.29) 18	0.00(0.00) 50	0
0.05	0.40(1.45) 46	0.04(0.01) 50	1.30(1.21) 19	0.04(0.01) 50	0.035
0.10	0.76(1.25) 37	0.14(0.03) 50	1.37(1.16) 18	0.21(0.34) 48	0.140
0.15	1.78(1.81) 23	0.32(0.06) 50	1.57(1.09) 16	0.43(0.42) 46	0.315
0.20	2.98(2.16) 12	0.57(0.10) 50	1.90(1.18) 15	0.72(0.53) 44	0.560
random	5.22(1.55) 0	2.40(0.87) 50	2.91(0.88) 12	2.76(1.01) 18	-

n = 10

ε	Method (a) Mean(S.D.) k	Method (b) Mean(S.D.) k	Method (c) Mean(S.D.) k	Method (d) Mean(S.D.) k	$E(n, \varepsilon)$
0.00	0.00(0.00) 50	- (-) -	4.01(2.61) 11	0.00(0.00) 50	0
0.05	0.57(2.19) -	- (-) -	3.79(2.71) -	0.07(0.01) -	0.075
0.10	1.91(3.10) -	- (-) -	4.32(2.80) -	0.29(0.04) -	0.300
0.15	5.93(4.98) -	- (-) -	4.97(2.62) -	1.09(1.46) -	0.675
0.20	9.87(5.20) -	- (-) -	5.52(2.41) -	2.24(1.84) -	1.200
random	14.17(2.59) -	- (-) -	6.67(1.04) -	6.65(1.11) -	-

[a] n and ϵ are the size of graphs and the noise level, respectively. Mean and S.D. give the sample means and the standard deviations of the criterion values $J(P)$, and k is the number of real optimum matchings obtained by each method. The expected value of the criterion is also given as $E(n, \epsilon)$.

TABLE II
PERFORMANCE EVALUATION OF THE WEIGHTED DIRECTED GRAPH MATCHING ALGORITHM (50 TRIALS PER DATUM)[a]

n = 7

ε	Method (a) Mean(S.D.) k	Method (b) Mean(S.D.) k	Method (c) Mean(S.D.) k	Method (d) Mean(S.D.) k	$E(n, \varepsilon)$
0.00	0.00(0.00) 50	0.00(0.00) 50	1.94(1.96) 24	0.00(0.00) 50	0
0.05	0.04(0.00) 50	0.04(0.00) 50	2.05(1.96) 23	0.04(0.00) 50	0.035
0.10	0.37(1.00) 47	0.14(0.02) 50	2.44(1.91) 19	0.14(0.02) 50	0.140
0.15	1.15(1.78) 39	0.32(0.04) 50	2.73(1.86) 17	0.32(0.04) 50	0.315
0.20	2.68(2.82) 28	0.57(0.08) 50	2.76(1.81) 19	0.77(0.82) 47	0.560
random	6.67(1.21) 0	3.39(0.49) 50	3.99(0.69) 5	3.92(0.66) 13	-

n = 10

ε	Method (a) Mean(S.D.) k	Method (b) Mean(S.D.) k	Method (c) Mean(S.D.) k	Method (d) Mean(S.D.) k	$E(n, \varepsilon)$
0.00	0.00(0.00) 50	- (-) -	4.71(4.28) 22	0.00(0.00) 50	0
0.05	0.08(0.01) -	- (-) -	4.79(4.28) -	0.08(0.01) -	0.075
0.10	1.06(2.11) -	- (-) -	5.25(4.16) -	0.31(0.03) -	0.300
0.15	4.81(4.72) -	- (-) -	6.10(4.09) -	0.69(0.07) -	0.675
0.20	8.48(5.24) -	- (-) -	6.92(3.83) -	1.68(1.78) -	1.200
random	14.41(1.79) -	- (-) -	9.02(1.00) -	8.95(1.05) -	-

[a] n and ϵ are the size of graphs and the noise level, respectively. Mean and S.D. give the sample means and the standard deviations of the criterion values $J(P)$, and k is the number of real optimum matchings obtained by each method. The expected value of the criterion is also given as $E(n, \epsilon)$.

not sufficiently close to each other ($\epsilon > 0.1$), this method usually fails to give good matchings. Method d), i.e., the hill-climbing method with an initial correspondence obtained by the proposed matching method, shows the best performance among the four methods. This method almost always succeeds in giving the real optimum matchings, even when the noise level is fairly high. This is because the initial correspondence obtained by the proposed method is considered to reflect the global correspondence between graphs.

TABLE III
CPU TIME ELAPSED (MILLISECONDS) DURING THE UNDIRECTED GRAPH MATCHING PROCESS (50 TRIALS PER DATUM)[a]

$\varepsilon = 0.10$

n	Method (a) Mean(S.D.)	Method (b) Mean(S.D.)	Method (c) Mean(S.D.)	Method (d) Mean(S.D.)
5	36(5)	142(21)	53(16)	56(9)
6	51(6)	491(109)	106(30)	84(17)
7	72(8)	1756(356)	213(56)	135(24)
8	98(7)	- (-)	347(77)	184(41)
9	132(10)	- (-)	549(127)	264(74)
10	172(16)	- (-)	851(276)	375(138)

$\varepsilon = 0.20$

n	Method (a) Mean(S.D.)	Method (b) Mean(S.D.)	Method (c) Mean(S.D.)	Method (d) Mean(S.D.)
5	36(8)	193(40)	53(17)	63(13)
6	54(7)	819(181)	105(29)	108(29)
7	76(7)	2841(429)	209(65)	173(41)
8	104(9)	- (-)	357(90)	282(97)
9	140(11)	- (-)	557(139)	417(142)
10	181(14)	- (-)	810(250)	665(241)

[a] n and ϵ are the size of graphs and the noise level, respectively. Mean and S.D. are the sample means and standard deviations of the total CPU time.

TABLE IV
CPU TIME ELAPSED (MILLISECONDS) DURING THE DIRECTED GRAPH MATCHING PROCESS (50 TRIALS PER DATUM)[a]

$\varepsilon = 0.10$

n	Method (a) Mean(S.D.)	Method (b) Mean(S.D.)	Method (c) Mean(S.D.)	Method (d) Mean(S.D.)
5	59(6)	158(7)	88(30)	88(12)
6	91(6)	461(12)	173(37)	142(20)
7	124(9)	1185(19)	317(94)	204(17)
8	173(9)	- (-)	589(147)	307(66)
9	233(12)	- (-)	978(326)	420(66)
10	300(16)	- (-)	1556(477)	558(91)

$\varepsilon = 0.20$

n	Method (a) Mean(S.D.)	Method (b) Mean(S.D.)	Method (c) Mean(S.D.)	Method (d) Mean(S.D.)
5	61(6)	177(28)	83(27)	98(15)
6	89(6)	562(100)	176(48)	167(43)
7	131(12)	1943(430)	319(99)	263(76)
8	180(12)	- (-)	602(184)	393(107)
9	240(12)	- (-)	936(267)	692(266)
10	316(19)	- (-)	1422(418)	1017(417)

[a] n and ϵ are the size of graphs and the noise level, respectively. Mean and S.D. are the sample means and standard deviations of the total CPU time.

VI. CONCLUSION

The approximate solution for the WGMP has been given in both the undirected and directed cases. The proposed method employs an analytic approach based on the eigendecomposition of the adjacency matrix of a graph and almost always gives the optimum matching when a pair of graphs are nearly isomorphic. If graphs are not sufficiently close to each other, the proposed method sometimes fails to give the optimum matching. However, the hill-climbing method can improve the obtained matching even in this case since the obtained matching reflects the global correspondence between graphs. This method is resistant to the combinatorial explosion in execution time compared to the purely combinatorial approach. It also has the virtue of global optimization, unlike the local optimization method.

When the weighted graphs are far different from the isomorphic cases, the method may not work as well as in the nearly isomorphic cases. However, it is usually meaningless to find the optimum matching between such graphs. Thus, the proposed method is considered to be satisfactory in most cases.

The proposed matching method requires a pair of graphs of the same size. This is an important restriction in practice since it means, for example, in a computer vision system, that the whole object of interest is in the field of view and that the segmentation and feature extraction process will always yield the correct results. It is difficult to satisfy these requirements in general. However, in the controlled world of industrial robot vision, they are sometimes satisfied [2]. Thus, the proposed matching method should be applied to such fields. Of course, the WGMP for the graphs with different numbers of nodes (the subgraph isomorphism problem in graph theory) is a more practical and important problem. However, this is left open for future work.

References

[1] W. H. Tsai and K. S. Fu, "Error-correcting isomorphisms of attributed relational graphs for pattern analysis," *IEEE Trans. Syst., Man, Cybernet.*, vol. SMC-9, pp. 757–768, Dec. 1979.

[2] H. S. Baird, *Model-Based Image Matching Using Location.* Cambridge, MA: M.I.T. Press, 1985.

[3] M. You and A. K. C. Wong, "An algorithm for graph optimal isomorphism," in *Proc. 1984 ICPR*, pp. 316–319, 1984.

[4] L. Kitchen and A. Rosenfeld, "Discrete relaxation for matching relational structures," *IEEE Trans. Syst., Man, Cybernet.*, vol. SMC-9, pp. 869–874, Dec. 1979.

[5] L. Kitchen, "Relaxation applied to matching quantitative relational structures," *IEEE Trans. Syst., Man, Cybernet.*, vol. SMC-10, pp. 96–101, Feb. 1980.

[6] M. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco, CA: W. H. Freeman, 1979.

[7] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications.* New York: Academic, 1980.

[8] A. J. Hoffman and H. W. Wielandt, "The variation of the spectrum of a normal matrix," *Duke Math. J.*, vol. 20, pp. 37–39, 1953.

[9] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithm and Complexity.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

Shinji Umeyama was born on August 25, 1954 in Tokushima, Japan. He received the B.S. and M.S. degrees in electrical engineering from Kyoto University in 1977 and 1979, respectively.

Since 1979, he has been with the Electrotechnical Laboratory, Ibaraki, Japan. From 1986 to 1987 he was a Visiting Research Scientist in the Division of Electrical Engineering of the National Research Council of Canada. His current research interests include graph theory, structural pattern recognition, and computer vision.