

A New Error-Correcting Distance for Attributed Relational Graph Problems

Yasser El-Sonbaty¹ and M. A. Ismail²

¹Dept. of Computer Eng., Arab Academy for Science & Tech., Alexandria 1029, EGYPT
yasser@aast.edu

²Dept. of Computer Sc., Faculty of Engineering, Alexandria 21544, EGYPT
mismail@alex.eun.eg

Abstract: In this paper a new distance for attributed relational graphs is proposed. The main idea of the new algorithm is to decompose the graphs to be matched into smaller subgraphs. The matching process is then done at the level of the decomposed subgraphs based on the concept of error-correcting transformations. The distance between two graphs is found to be the minimum of a weighted bipartite graph constructed from the decomposed subgraphs. The average computational complexity of the proposed distance is found to be $O(N^4)$, which is much better than many techniques.

1- Introduction

The search for general structural mathematical models has led workers in the field of pattern recognition to study graphs, for these can be of direct use in describing relations between the elements of a set of objects. Further, graph theory methods can be used in a wide variety of problems and for this reason much study has been given to the mathematical and algorithmic properties of graphs.

Attributed relational graphs *ARGs* are one of the most powerful tools in describing structured objects. In this representation, nodes represent primitives or subpatterns of structured objects and branches between nodes represent relations between primitives or subpatterns [1].

One way to recognize the structure of an unknown pattern is to transform it into an *ARG*, then match it with other *ARGs* representing structures of prototype patterns. This process of matching is called graph isomorphism. Formally, two graphs G and G' are said to be isomorphic (to each other) if there is a one-to-one correspondence between their vertices and between their edges such that incidence relationship is preserved [2]. If the isomorphism is encountered between a graph and a subgraph of another larger graph, then the problem is called subgraph isomorphism or graph monomorphism. Largest common subgraphs problem is to find an isomorphic mapping between subgraphs of G and subgraphs of G' .

Graph isomorphism was widely used as a powerful tool for matching and recognizing structured objects using different techniques like : inexact graph matching [4], relaxation methods [5], Cartesian graph product [6,17], error-correcting transformations [7,14], neural networks [8,18], graduate assignment [9] and direct

classification of node attendance [10]. Trials for matching weighted graphs were shown in [11-13]. Some of the applications were demonstrated in [19, 27].

The main problem of using graph isomorphism as a tool for graph matching that it is only permitted when matched graphs have some common structures and that means, graph isomorphism can not be used when matching graphs with different structures [21]. In this case, a measure for distance between graphs is needed. Some contributions were recognized in introducing efficient distances between graphs as in [1, 22-30] using three general methods based on 1- feature extraction [20], 2- graph grammar [22-24], and 3- error-correcting transformations [1, 25-30]. The main problems of these distance measures are the complexity which may grow up exponentially when increasing the sizes of matched graphs and their deficiency in handling graph isomorphism problems.

In this paper a distance measure between attributed relational graphs is introduced. The proposed distance can be efficiently used for determining the isomorphism between matched graphs. The basic idea of the proposed algorithm is to decompose the matched graphs into smaller subgraphs and perform the matching between the graphs at the level of their decomposed subgraphs based on the concept of error-correcting transformations.

The process of graph decomposition, and how to match the decomposed subgraphs are shown in section 2. Section 3 introduces the proposed algorithm for calculating the distance between matched graphs with analysis of its computational complexity. Experimental results are presented in section 4, and finally the conclusions of the proposed algorithm are given in section 5.

2- Graph Decomposition

In this section the process of graph decomposition into smaller subgraphs is introduced, followed by proposing the matching algorithm of these decomposed subgraphs.

2-1- Decomposition of Attributed Relational Graphs.

Simplifying the structure of matched graphs will certainly reduce the overall complexity of an algorithm that enhances the performance.

The graphs resulting from decomposing an ARG are called *Basic Attributed Relational Graphs (BARGs)* and this notion is adopted from [27]. The BARG is in the form of one level tree which consists of a node and all nodes connected to that node whether the connected branches are in or out of that node. The structure of these BARGs gives the matching process more stability, increases the data associated with decomposition process, facilitates the matching process, and preserves the structure of the original graph.

2-2- Matching BARGs

Matching BARGs is much easier than matching ARGs and this is because the structure of BARGs is simpler and easier in processing than ARGs. The same distance measures for ARGs can also be adopted for BARGs. In this paper, we use the concept

of error-correcting transformations, where the cost of matching two ARGs is defined as the cost of the sequence of transformations that possesses minimum total cost and that must be performed on one of the two ARGs in order to produce the other ARG. The operations permitted to transform one ARG to the other are : *node insertion, node deletion, branch insertion, branch deletion, node label substitution* and *branch label substitution*. A cost (weight) is associated with each operation and its value is determined by an optimization procedure or heuristically. The costs corresponding to each operation are : w_{ni} , w_{nd} , w_{bi} , w_{bd} , w_{ns} , and w_{bs} respectively. Some research in this area can be found in [1,7, 14, 25-28].

A new operation *structure preservation* is added here with cost w_{sp} . The main function of the new operation is to help preserving the global structure of the original ARG after performing the operation of graph decomposition and its use will be declared later in this section.

Given two BARGs, say U and V as shown in Fig. 1. the cost of matching U and V is calculated as follows :

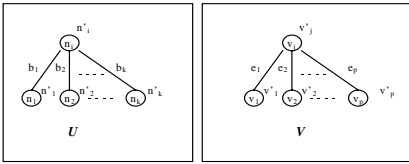


Fig. 1. Two BARGs; U and V

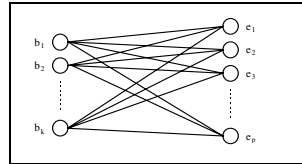


Fig. 2. corresponding weighted bipartite graph

$$\text{Distance}(U, V) = w_{ns} * \text{dist}(n_i, v_j) + \min(w_{ni}, w_{nd}) * \text{abs}(k-p) + \min(w_{bi}, w_{bd}) * \text{abs}(k-p) + \text{dist}(b's, e's) \tag{1}$$

where, $\text{dist}(n_i, v_j)$ = distance between node labels of n_i and v_j and node attributes of n'_i and v'_j and is calculated depending on their data types. k and p are the number of branches connected to the root node of the matched BARGs.

$\text{dist}(b's, e's)$ is calculated as the minimum of a weighted bipartite graph constructed from $b's$ and $e's$ as their nodes and its structure is as shown in Fig. 2.

the weight of any branch connecting two nodes, say b_f and e_h in the bipartite graph is the distance between b_f and e_h and is calculated as follows :

$$\text{distance}(b_f, e_h) = w_{bs} * \text{dist}(b_f, e_h) + w_{sp} * \text{dist}(n_f, v_h) \tag{2}$$

where $\text{dist}(x,y)$ is the distance between x and y and their attributes based on their data type.

The minimum of the weighted bipartite graph can be calculated by many algorithms [31-33] and is known in the literature as assignment problem and it has computational complexity $O(n^3)$ in the worst case and $O(n^2)$ in the average case[33].

3- The Algorithm

The main idea of the proposed algorithm is to decompose both reference and input graphs; say G_R and G_i respectively; to be matched into $BARGs$ as previously mentioned in section 2-1. A distance matrix “ D ” between reference and input graphs is then constructed and is equivalent to the distances between the $BARGs$ of both reference and input graphs. The labels of the rows in the distance matrix represent the $BARGs$ of the input graph, while the labels of the columns represent the $BARGs$ of the reference graph. $D_{i,j}$ represents the distance between the i^{th} $BARG$ in the input graph and the j^{th} $BARG$ in the reference graph and is calculated as mentioned in section 2-2.

After calculating the distance matrix D , a weighted bipartite graph is constructed which is equivalent to the distance matrix D where each branch connecting two nodes, one represents a $BARG$ from the input graph and the other node represents a $BARG$ from the reference graph, has a weight equivalent to the distance between the two $BARGs$ connected by this branch as shown in Fig. 3.

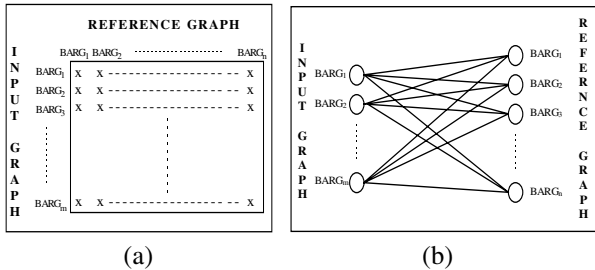


Fig. 3. (a) the distance matrix, (b) corresponding weighted bipartite graph matrix.

$$Distance(input_graph, reference_graph) = \text{minimum_weighted_bipartite_graph}(\text{distance matrix}) + \text{unmatched_branches} + \text{unmatched_nodes} \quad (3)$$

Every pair of $BARGs$ (one $BARG$ from input graph and the other is from reference graph) in the weighted bipartite graph connected by a branch whose weight is taken in calculating the minimum of the weighted bipartite graph, is considered to be matched, i.e., the node rooted the $BARG$ of input graph matches the node rooted the $BARG$ of reference graph. Fig 4-(a) shows the structure of minimum weighted bipartite graph while Fig. 4-(b) shows the position of unmatched branched and nodes.

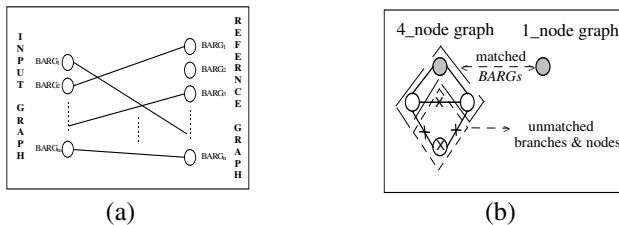


Fig. 4. (a) Min. weighted bipartite graph, (b) Unmatched branches and nodes

3-1- Analysis of Computational Complexity

Suppose that we are given a reference graph R with M nodes and an input graph I with N nodes. The first step in the algorithm is to decompose both R and I into $BARGs$. It is obvious that this step is done in quadratic time for each graph, i.e., has time complexity of $O(M^2+N^2)$.

The second step in the algorithm is to calculate the distance between all the $BARGs$ of input and reference graphs. The computational complexity of matching two $BARGs$ mainly depends on the calculation of error-transformations between the two $BARGs$ and the computational complexity of getting the minimum of a weighted bipartite graph. The computational complexity of calculating the error-correcting transformations is $O(\delta^2)$, where δ denotes the maximum number of branches connected to any node in the matched $BARGs$. The average computational complexity of getting the minimum of a weighted bipartite graph is $O(M*N)$ [48], assuming that any node is connected to all other nodes in the graph. so we can consider the computational complexity of matching two $BARGs$ is $O(M*N)$. The calculation of the distance between the $BARGs$ is repeated for all the $BARGs$ of both input and reference graphs, and that means, the average computational complexity of second step is $O(M^2*N^2)$, where $M*N \geq \delta$. The best and worst cases for this step have computational complexity of $O(M*N*min(M,N))$ and $O(M^2*N^2*min(M,N))$ respectively.

The third step is to calculate the cost of matching input and reference graphs which is defined as the minimum of the weighted bipartite graph constructed in step 2 and the computational complexity of this step is $O(M*N)$.

The last step is to count number of unmatched branches and unmatched nodes. This step has complexity of quadratic order.

In summary the average computational complexity of the proposed algorithm in calculating the cost of matching two $ARGs$ with M and N nodes is $O(M^2*N^2)$. The best and worst computational complexity are $O(M*N*min(M,N))$ and $O(M^2*N^2*min(M,N))$ respectively. The computational complexity of the new algorithm is much better than other algorithms reported. in the literature [1, 7, 9, 10, 14, 25-30].

4- Experimental Results

4-1- Distance between Graphs :

We start by demonstrating the capability of the new algorithm in calculating the distance between attributed relational graphs.

The problem is to identify an image graph of the prominent runways of the Jacksonville airport from the picture shown in Fig. 5.[3]. The image graph has to be matched with three runway models of the open-V runway configuration. Airports that have open-V runway are Houston airport, Jacksonville airport, and finally Mid-continent airport and are shown in Fig 6- (a), (b), and (c) respectively. A vertex represents a runway and has an attribute that corresponds to the length of this runway.

Each edge has two attributes; center to center distance between connected vertices and cute angle between them [3].

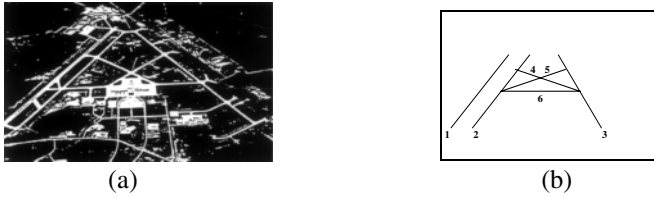


Fig. 5. (a) Jacksonville airport [3], (b) Image graph of Jacksonville airport.

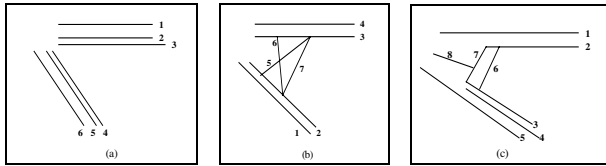


Fig. 6. (a) Runway model A., (b) Runway model B., (c) Runway model C

Table I shows the output of the proposed algorithm, which indicates that the image graph of the Jacksonville runways and runway model B incur the lowest cost of matching. The results of matching are consistent and identical with the results published in [3].

Base Graph	Matching Cost	Vertex Mapping
Model A	5660	(1,6),(2,5),(3,1),(4,2),(5,4),(6,3))
Model B	3847	(1,1),(2,2),(3,4),(4,5),(5,6),(6,7))
Model C	4430	(1,5),(2,4),(3,1),(4,2),(5,3),(6,8))

Table I. Results of subgraph optimal isomorphism.

4-2- Graph Isomorphism Problem :

In this experiment, the performance of the proposed algorithm is tested in matching sparsely attributed relational graphs in different noise levels. Attributed relational graphs of 50 nodes are generated with degree of connectivity γ , where $\gamma \in \{10\%, 15\%, 20\%, 25\%\}$. Two nodes are connected by an edge with probability γ . Different noise levels are added to edge and node attributes. Noise levels are in the range of $\{0.00, 0.04, 0.08, 0.12, 0.16, 0.20\}$. After adding the noise, the graph is shuffled and matched with the original one. One hundred trial is produced for each connectivity and noise level. Fig. 7 shows the results obtained from applying the proposed algorithm.

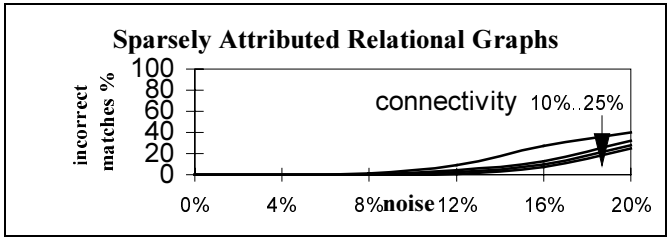


Fig. 7. Results of experiment 2 for sparsely attributed relational graphs.

4-3- Subgraph Isomorphism Problem :

In this section the performance of the proposed algorithm is tested in handling the problem of subgraph isomorphism.

In this experiment we use simulation method to test the performance of the new algorithm in handling sparsely connected graphs. Attributed relational graphs of size 100 are generated with 10% degree of connectivity. The weights of edges are real numbers and are produced randomly in the rang of 0 - 1. Nodes have five random binary valued attributes. After the generation of the graphs, 60% or 80% nodes are deleted and uniform noise is added to the edges. The noise levels are in the rang of {0.00, 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18, 0.20}. One hundred trials are run at each noise level. The results of the proposed algorithm in comparison with the results obtained from the algorithm of graduated assignment [9] are shown in figures 8-9 where (a) is the proposed algorithm and (b) is the graduated assignment algorithm. From figures 8-9, it can be concluded that the proposed algorithm has lower percentage of incorrect matches than the recent technique of graduated assignment [9].

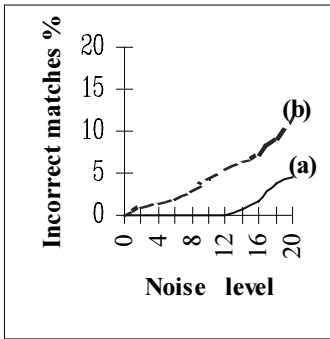


Fig. 8 Results of matching ARGs with 5 binary attributes and 60% deleted.

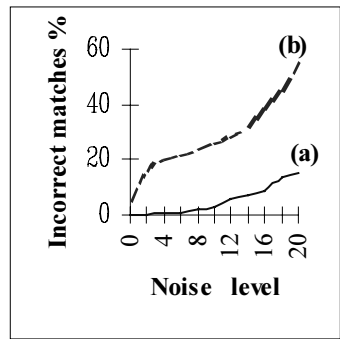


Fig. 9 Results of matching ARGs with 5 binary attributes and 80% deleted.

4-4- Largest Common Subgraphs Problem:

Finally, the new algorithm is tested in handling the problem of largest common subgraphs. A modification is done in the proposed algorithm that some of the entries in the results of the minimum weighted bipartite graph are excluded because their

values (which are equivalent to dissimilarity distance between some *BARGs*) are higher than some pre-defined threshold value " t_{iso} ". The threshold value is application dependent and is determined by an optimization procedure or heuristically. The main purpose of t_{iso} is to ensure that only isomorphic nodes are included. Fig. 10 shows the distribution of *BARGs* construction the minimum weighted bipartite graph.

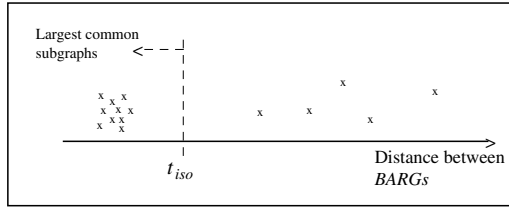


Fig. 10. Distribution of *BARGs* and expected position of t_{iso} .

Fig. 11 and Fig. 12 show different scans for an autonomous robot [17]. The data contained in each scan is represented by attributed relational graph where vertices are boundaries and classified to four categories : wall, pseudo-boundary, partial-wall, and reference-boundary. The length of each boundary is added as a vertex attribute. The edges in the graph represent the angles between the boundaries. Table II. shows the results obtained from the proposed algorithm on this data. The results are completely consistent and identical with those in [17] . The algorithm of [17] was reported to be exponential.

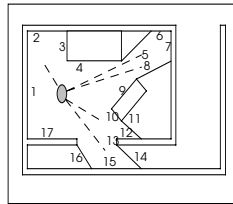


Fig. 11. Scan 1

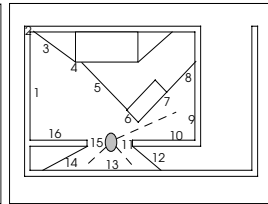


Fig. 12. Scan 2

Vertex Mapping		Edge Mapping	
Scan 1	Scan 2	Scan 1	Scan 2
1	1	(1-17)	(1-16)
10	6	(17-1)	(16-1)
17	16	-	-

Table II. Results of Exp. 1.

5- Conclusions

In this paper, a new algorithm for evaluating the distance between attributed relational graphs is proposed. From experimental results and complexity analysis, the following points can be concluded:

- 1- the new algorithm can be used efficiently for sparsely and fully connected attributed relational graphs, and also for other types of graphs like attributed graphs and weighted graphs
- 2- the new algorithm has the capability of handling different isomorphism problems like graph isomorphism, subgraph isomorphism and largest common subgraphs with distinguished performance,
- 3- the computational complexity of this algorithm is much lower than other techniques found in the literature. The average computational complexity of the proposed algorithm is found to be $O(M^2*N^2)$,
- 4- the best and worst case for the computational complexity of the new algorithm is $O(M*N*min(M,N))$ and $O(M^2*N^2*min(M,N))$ respectively, which is better than many techniques,
- 5- the proposed algorithm is based on the concept of error-correcting transformations which is more powerful and has more meaning in calculating the distance than other techniques.
- 6- the proposed algorithm is parallel in nature and can take advantage of hardware parallel architecture.

References

- 1- Sanfeliu A., Fu K. S.: A Distance Between Attributed Relational Graphs for Pattern Recognition, IEEE Trans. on Sys., Man and Cybernetics, Vol. 13, No. 3, (1983) 353- 362.
- 2- El-Sonbaty Yasser, Ismail, M. A.: A New Algorithm for Subgraph Optimal Isomorphism, Pattern Recognition, Vol. 31, No. 2. (1998) 205-218.
- 3- Wong, A. K. C., You, M, Chan, S. C.: An Algorithm for Graph Optimal Monomorphism, IEEE Tran. on Sys., Man and Cybernetics, Vol. 20, No. 3, (1990) 628-636.
- 4- Shapiro, L. G. Haralick, R. M.: Structural Descriptions and Inexact Matching, Tech. Rep. CS-79011-R, Virginia Polytech. Inst. and State Univ., Blacksburg VA, (Nov. 1979).
- 5- Kitchen, L.: Relaxation Applied to Matching Quantitative Relational Structures, IEEE Tran. on Sys., Man and Cybernetics, Vol. 10, No. 2, (1980).
- 6- Ghahraman, D. E., Wong, A. K. C., Au, T.: Graph Optimal Monomorphism Algorithms, IEEE Tran. on Sys., Man and Cybernetics, Vol. 10, No. 4, (1980).
- 7- Tsai, W., Fu, K. S.: Subgraph Error-Correcting Isomorphisms for Syntactic Pattern Recognition, IEEE Tran. on Sys., Man and Cybernetics, Vol. 13, No. 1, (1983) 48-62.

- 8- Suganthan, P., Teoh, E., Mital, D.: Pattern Recognition by Graph Matching Using the Potts MFT Neural Networks Pattern Recognition, Vol. 28, (1995) 997-1009.
- 9- Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching, IEEE Trans. on PAMI, Vol. 18, No. 4, (1996) 377-388.
- 10- Depiero, F., Trivedi, M., Serbin, S.: Graph Matching Using a Direct Classification of Node Attendance Pattern Recognition, Vol. 29, No. 6, (1996) 1031-1048.
- 11- Umeyama, S.: An Eigendecomposition Approach to Weighted Graph Matching Problems, IEEE Trans. on PAMI, Vol. 10, No. 5, (1988) 695-703.
- 12- Almohamad, H. A.: A Polynomial Transform for Matching Pairs of Weighted Graphs, J. Applied Math. Modeling, Vol. 15, No. 4.,(1991).
- 13- Almohamad, H. A., Duffuaa, S. O.: A Linear Programming Approach for the Weighted Graph Matching Problem, IEEE Trans. on PAMI, Vol 15, No. 5.(1993).
- 14- Tsai W. H., Fu K. S.: Error-Correcting Isomorphism of Attributed Relational Graphs for Pattern Rec., IEEE Trans. on Sys., Man, and Cybernetics, Vol. 9, No. 12, (1979) 757-768.
- 15- Schoning U.: Graph Isomorphism is in the Low Hierarchy, J. of Computer and System Sciences, No. 37, (1988) 312-323.
- 16- You, M., Wong, A. K. C.: An Algorithm for Graph Optimal Isomorphism, in Proc. 1984 ICPR, (1984) 316-319.
- 17- Stacey D. A., Wong, A. K. C.: A Largest Common Subgraph Isomorphism Algorithm for Attributed Graphs, Tech. Report, Dept. of System Design Eng., U. of Waterloo, Waterloo, ONT, June 1988.
- 18- Shoukry A., Aboutabl M.: Neural Network Approach for Solving the Maximal Common Subgraph Problem, IEEE Trans. on Sys., Man, and Cyber., Vol. 26, No. 5, (Oct. 1996).
- 19- Bayada D. M., Simpson R. W., Johnson A. P.: An Algorithm for the Multiple Common Subgraph Problem, J. Chem. Inf. Comput. Sci., Vol. 32, (1992) 680-685.
- 20- Lowe, A.: Three Dimension Object Recognition from Single Two Dimension Images, Artificial Int., Vol. 31,(1987) 335-395.
- 21- Schalkoff R.: Pattern Recognition: Statistical, Structural and Neural Approaches, John Wiley & Sons Inc., (1992).
- 22- Shaw A. C.: A Formal Picture Description Scheme or a Basis for Picture Processing Systems, Information and Control, Vol. 14, (1969) 9-52.
- 23- Pavlidis T.: Linear and Context-Free Graph Grammars, JACM, Vol. 19, (1972) 11-22.
- 24- Tanaka E.: Theoretical Aspects of Syntactic Pattern Recognition, Pattern Recognition, Vol. 28, No. 7, (1995) 1053-1061.
- 25- Kajitani Y., Sakurai H.: On Distance of Graphs Defined by Use of Orthogonality Between Circuits and Cutsets Rep. CT-73-30, The Inst. of Elect. and Comm., Engineers of Japan, (1973).
- 26- Tanaka E.: A Metric on Graphs and its Applications, IEE Japan ip-77-55, OCT. 31, (1977).
- 27- Eshera M. A., Fu K. S.: An Image Understanding System Using Attributed Symbolic Representation and Inexact Graph Matching, PAMI, Vol. 8, No. 5, (1986) 604-617.

- 28- Eshera M. A., Fu K. S.: A Graph Distance Measure for Image Analysis, IEEE Tran. on Sys., Man and Cybernetics, Vol. 14, No. 3, (1984) 398-408.
- 29- Shapiro L., Haralick R.: A Metric for Relational Descriptions, IEEE Trans. Pattern Analysis and Mach. Int., 7, (1985) 90-94.
- 30- Cinque L., Yasuda D., Shapiro L. G., Tanimoto S., Allen B.: An Improved Algorithm for Relational Distance Graph Matching, Pattern Recognition, Vol. 29, No. 2, (1996) 349-359.
- 31- Kuhn, H. W.: The Hungarian Method for the Assignment Problem, Naval Res. Log. Quart., No. 12, (1955) 83-97.
- 32- Karp, R. M.: An Algorithm to Solve the $m*n$ Assignment Problem in expected time $O(mn \log n)$, Networks, No. 10, (1980) 143-152.
- 33- Yamada, S., Kasai, T.: An Efficient Algorithm for the Linear Assignment Problem, Elect. and Comm. in Japan, Part 3, Vol. 73, No. 12, (1990) 28-36.