http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (14): 2969-2974, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.2969.2974

ASE Approach for Large Graphs Matching

[1]Anliang Ning, [2]Xiaojing Li and [3]Chunxian Wang
[1]Engineering Training Center, Tianjin Polytechnic University, Tianjin 300387, China
[2]College of Electronic Engineering, Tianjin Polytechnic University, Tianjin 300387, China
[3]College of Computer Sciences and Software, Tianjin Polytechnic University, Tianjin, China

Abstract: How to match two large graphs by maximizing the number of matched edges, which is known as maximum common subgraph matching and is NP-hard. The Anchor-Selection and Expansion (ASE) approach to compute an initial matching is presented in the study. We give heuristics to select a small number of important anchors using a new similarity score, which measures how two nodes in two different graphs are similar to be matched by taking both global and local information of nodes into consideration. And then by expanding from the anchors selected we work out a good initial matching. The expansion is based on structural similarity among the neighbors of nodes in two graphs. The approach that can efficiently match two large graphs over thousands of nodes with high matching quality is proved in theorized.

Key words: Large graph match, maximum common subgraph, global node similarity, anchor selection and expansion

INTRODUCTION

Graph proliferates in a wide variety of applications, including social networks in psycho-sociology, attributed graphs in image processing, food chains in ecology, electrical circuits in electricity, road networks in transport, protein interaction networks in biology, topological networks on the Web. Graph processing has attracted great attention from both research and industrial communities. Graph matching is an important type of graph processing, which aims at finding correspondences between the nodes/edges of two graphs to ensure that some substructures in one graph are mapped to similar substructures in the other. Graph matching plays an essential role in a large number of concrete applications.

The graph matching literature is extensive and many different types of approaches have been proposed, which mainly focus on approximations and heuristics for the quadratic assignment problem. An incomplete list includes spectral methods, relaxation labeling and probabilistic approaches, semi-definite relaxations, replicator equations, tree search, graduated assignment and RKHS methods (Plantenga, 2013). A number of algorithms have been proposed for graph matching including exact matching (Egozi *et al*., 2013) and approximate matching (Plantenga, 2013). The exact approaches are able to find the optimal matching at the cost of exponential running time, while the approximate approaches are much more efficient but can get poor matching results. More importantly, most of them can only handle small graphs with tens to hundreds of nodes. As an indication, exactly matching two undirected graphs with 30 nodes may take time about 100,000s. It is important to note that real-world networks nowadays can be very large. The existing approaches cannot efficiently match graphs even with thousands of nodes with high quality.

In this study, we study the problem of matching two large graphs, which is formulated as follows. Given two graphs G1 and G2, we find a one-to-one matching between the nodes in G1 and G2 such that the number of the matched edges is maximized. The optimal solution to the problem corresponds to the Maximum Common Subgraph (MCS) between G1 and G2, which is an NP-hard problem and has been studied in decades. It is known to be very difficult to find a high-quality approximate matching efficiently even for small graphs. In order to meet the needs of handling large graphs for graph matching and analysis, we propose a novel approximate solution with polynomial time complexity while still attaining high matching quality. The rest of the study is organized as follows. Section 2 gives the problem statement. Section 3 gives the anchor-selection/expansion approach and its application examples. Section 4 concludes this study.

PROBLEM STATEMENT

We first focus on undirected and unlabeled graphs, since the most difficult part for graph matching is the structural matching without any assistance of labels. We

Corresponding Author: Anliang Ning, Engineering Training Center, Tianjin Polytechnic University, Tianjin 300387, China

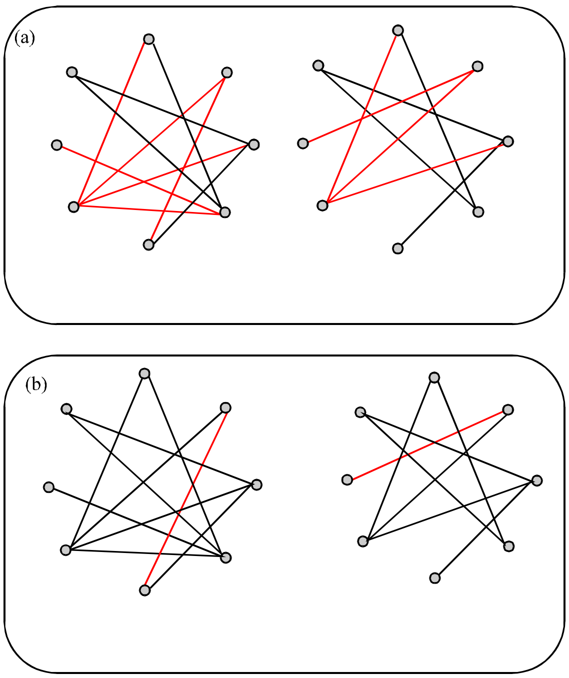

Fig. 1(a-b): (a) MCSv and (b) MCSe

will discuss how to handle labeled graphs later in this study. For a graph G(V, E), we use V(G) to denote the set of nodes and E(G) to denote the set of edges.

Definition 1: Graph/Subgraph Isomorphism: Graph G1 is isomorphic to graph G2, if and only if there exists a bijective function f: V(G1)→V(G2) such that for any two nodes u1□V(G1) and u2□V(G1), (u1, u2)□E(G1) if and only if (f(u1), f(u2))□E(G2). G1 is subgraph isomorphic to G2, if and only if there exists a subgraph G' of G2 such that G1 is isomorphic to G'.

Definition 2: Maximum common subgraph: A graph G is the maximum Common Subgraph (MCS) of two graphs G1 andG2, denoted as mcs (G1, G2), if G is a common subgraph of G1 and G2 and there is no other common subgraph G', such that G' is larger than G.

The MCS of two graphs can be disconnected and there are two kinds of MCSs, namely maximum common node induced subgraph (MCSv) and maximum common edge induced subgraph (MCSe). The former requires the MCS to be the node induced subgraph of both G1 and G2 and G' is larger than G iff |V(G')|>|V(G)|. The latter requires the MCS to be the edge induced subgraph of both G1 and G2 and G'is larger than G iff |E(G')|>|E(G)|. Figure 1 shows the difference between MCSv and MCSe. Figure 1a shows the MCSv of G1 and G2, whereas Fig. 1b shows the MCSe of G1 and G2.

As can be seen from this example, MCSe can possibly get more common substructure for the given two graphs. In this study, we adopt MCSe since it can possibly get more common substructure for the given two graphs and we use MCS (mcs) to denote MCSe. Finding the MCS of two graphs is NP-hard.

Definition 3: Graph matching: Given two graphs G1 and G2, a matching M between G1 and G2 is a set of vertex pairs M ={(u,v)|u□V(G1), v□V(G2)}, such that for any two pairs (u1,v1) □ M and (u2,v2)□M, u1≠u2 and v1≠v2. The optimal matching M of two graphs is the one with the largest number of matched edges. Finding the optimal matching M is the same as finding the MCS.

Problem statement: We aim to compute the optimal matching M for two given graphs G1 and G2. For a given matching M, we evaluate its quality by computing score (M) as follows:

$$\text{Score (M)} \; \frac{\sum_{(u1,v1)\in M}\sum_{(u2,v2)\in M} e_{u1,u2}\times e_{v1,v2}}{2} \tag{1}$$

where $e_{u,v}$ = 1 if there is an edge between u and v and $e_{u,v}$ = 0, otherwise. Obviously, finding the optimal matching M is actually to find a matching with the maximum score (M) and the maximum score (M) is |E(mcs(G1, G2))|.

It is known that the MCS problem is NP-hard and it is also known that it is very difficult to obtain a tight, or even useful, approximation bound, because finding a maximum common subgraph of two graphs is equivalent to finding a maximum clique in their association graph, which cannot be approximated with ratio $n^{\varepsilon}$for any constant ε> 0 unless P = NP. For the quality of the MCS result, (Tang *et al*., 2012) give a bound of O(n2) based on the number of mismatched edges, where n is the size of the larger graph. This means that it may mismatch all the edges. Zhi-Yong *et al*. (2012) provide an upper bound for the size of the MCS, which is computed by sorting the degree sequences of two graphs separately followed by summarizing the corresponding smaller degrees. The bound is almost the smaller graph, without considering any structural information of the two graphs, which does not provide much information. For the time complexity, in (Kpodjedo *et al*., 2012), it is $O(n^6 L)$, where n is the size of the graph and L is the size of an LP model formulated for graph matching (at least n). It cannot handle graphs with more than 100 nodes.

ASE MATCHING APPROACH

In this study, we propose a novel approach to solve the graph matching problem. We construct the initial matching M by identifying anchors of two graphs G1 and G2 followed by expanding from the anchors. We do so based on a new similarity between nodes in the two different graphs, which combines both global and local information of nodes. The framework of the algorithm is shown in Algorithm 1.

Algorithm 1: Match (G1, G2)

Require: two graphs, G1 and G2;
Ensure: a graph matching between G1 and G2;
1: A←anchor-selection (G1, G2); {refer to Algorithm 2}
2: M←anchor-expansion (G1, G2, A); {refer to Algorithm 3}
3: M←refine(G1, G2, M);
4: return M;

In this section, we discuss how to select anchors and how to expand from the selected anchors to obtain the initial matching M for two graphs G1 and G2, using a new node similarity matrix S. The node similarity between u□G1 and v□G2 is very important because it indicates how likely the two nodes will be matched when computing the matching M.

Global and local node similarity: Let G1 and G2 be two graphs. The new node similarity matrix S we propose takes both global and local node similarities into consideration when matching nodes in two graphs:

$$S(u,v) = Sg(u,v)\times Sl\,(u,v) \tag{2}$$

Here, S is a |V(G1)|×|V(G2)| matrix, in which the element S[u,v]□[0, 1] represents the similarity of two nodes, u in G1 and v in G2. S is based on Sg and Sl, where Sg measures global similarity between u and v in the entire graphs G1 and G2 and Sl measures local similarity between u and v in their neighborhoods.

We will introduce an existing global similarity below followed by the discussion on our new local similarity in this section.

Global node similarity: In the literature, the global similarity for nodes in two graphs can be the spectral-based similarity. The representative study is Umeyama's work (Bhattacharjee and Jamil, 2012) which is improved by (Cheng *et al*., 2011). Suppose G1 and G2 are two undirected graphs with the same number of nodes n. The Laplacian matrix $L_{n\times n}$ of graph G with n nodes is defined as:

$$L = D\text{-}A$$

where, A is the adjacency matrix and D is the diagonal degree matrix. A[u1, u2] = 1 if (u1, u2) □ E(G) and 0 otherwise. D[u1, u1]= $\sum_{(u1,u2)?E(G)}$A[u1, u2]. We denote the Laplacian matrices of G1 and G2 as L1 and L2, respectively. Suppose the eigenvalues of L1 and L2 are $\alpha 1 \geq \alpha 2 \geq \ldots \geq \alpha n$ and $\beta 1 \geq \beta 2 \geq \cdots \geq \beta n$, respectively. Since L1 and L2 are symmetric and positive-semidefinite, we have $L1 = U1\Lambda 1U1^T$ and $L2 = U2\Lambda 2U2^T$, where U1 and U2 are orthogonal matrices and $\Lambda 1$ = diag (αi) and $\Lambda 2$ = diag (βi). If G1 and G2 are isomorphic, there exists a permutation matrix P such that $PU1\Lambda 1U1^TP^T = U2\Lambda 2U2^T$. Let P = U2D' $U1^T$ where D' = diag (d1,..., dn) and di □ {+1;-1} accounts for the sign ambiguity in the eigende-composition. When G1 and G2 are isomorphic, the optimum permutation matrix is P, which maximizes tr:

$$(P^T\overline{U_2}\,\overline{U_1^T})$$

where, $\overline{U_1}$ and $\overline{U_2}$ are matrices that have the absolute value of each element of U1 and U2, respectively. When the numbers of nodes in G1 and G2 are not the same, we only choose the largest c eigenvalues (Kpodjedo *et al*., 2010). Let c = min{|V(G1)|, |V(G2)|} and $\overline{U_1'}$ and $\overline{U_2'}$ be the first c columns of $\overline{U_1}$ and $\overline{U_2}$, respectively, the global similarity matrix can be computed with Eq. 3:

$$Sg = \overline{U_1'}\,\overline{U_2'}^T \tag{3}$$

Here, Sg[u,v]□[0, 1] is the global node similarity between the node u in V(G1) and the node v in V(G2). Example 1 shows an example of matching two graphs using the global node similarity.

Example 1: Consider the two graphs in Fig. 2. We first compute their global node similarity matrix Sg. We construct a bipartite graph Gb with |V(G1)|+|V(G2)| nodes and for any u□ V(G1) and v□ V(G2), we add an edge (u,v) □ E(Gb) with weight Sg[u,v].

We compute the maximum weighted bipartite matching of Gb and get the matching as M ={(u1,v1), (u2, v2), (u3, v7), (u4, v4), (u5, v5), (u6, v12), (u7, v13), (u8, v8), (u9, v17), (u10, v10), (u11, v3), (u12, v6), (u13, v14), (u14, v15), (u15, v16), (u16, v9)}. In this way, the number of matched edges is 10, which is far away from the optimal solution mcs (G1, G2), 21 (bold edges in Fig. 2). Comparing to the optimal solution, u3 is mismatched to v7 because they have a high global similarity, but obviously, the local structure near u3 and the local structure near v7 differ much.

Local node similarity: For any node v in graph G and k = 0, we define the k-neighborhood of v, $K_{k(v)}$, as the set

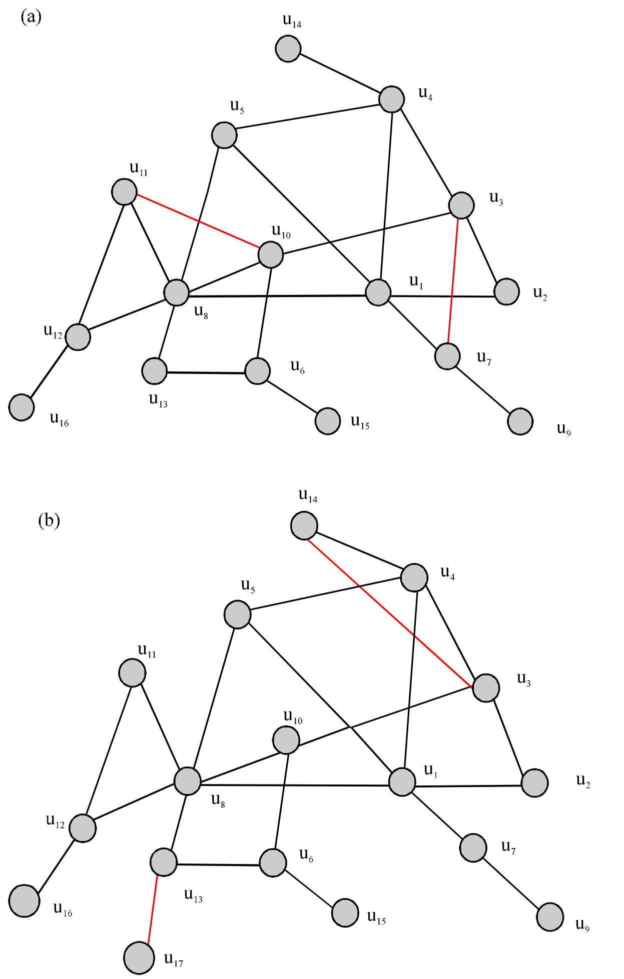

Fig. 2(a-b): Two graphs, (a) Graph G1 and (b) Graph G2

of nodes in V(G) such that v/□ $N_{k(v)}$ and for any u □$N_{k(v)}$, the shortest distance from v to u is no more than k. The shortest distance is defined as the number of edges in the shortest path from v to u. The k-neighborhood subgraph of v in G, denoted as G^k_v, is defined as the induced subgraph over $N_{k(v)}$ □ {v} in G. For two nodes u □ V (G1) and v □ V(G2), we measure their local node similarity by comparing the k-neighborhood subgraphs of them. Suppose d(u) and d(v) are the degrees of node u and v in G1 and G2, respectively and suppose d1,1, d1,2,... is the degree sequence of node set $N_{k(u)}$ in G^k_u sorted in non-increasing order and d2,1, d2,2,... is the degree sequence of node set $N_{k(u)}$ in G^k_u sorted in non-increasing order. Let nmin = min{$|N_{k(u)}|$, $|N_{k(v)}|$}.We define a |V(G1)|×|V(G2)| local node similarity matrix Sl as follows:

$$S_l[u,v]=\frac{(n_{min}+1+D(u,v))^2}{(|V(G^k_u)|+|E(G^k_u)|)(|V(G^k_v)|+|E(G^k_v)|)} \quad (4)$$

$$D[u,v]=\frac{\min\{d(u),d(v)\}+\sum_{i=1}^{n_{min}}\min\{d_{1,i},d_{2,i}\}}{2} \quad (5)$$

Here, D(u,v) consists of two parts. The first part min{d(u), d(v)} is the ideal contribution of edges when matching u with v and the second part:

$$\sum_{i=1}^{n_{min}}\min\{d_{1,i},d_{2,i}\}$$

is the ideal contribution of edges when matching nodes in $N_{k(u)}$with nodes in $N_{k(v)}$. We show that Sl has the following properties:

- <Sl [u,v] = 1
- $S_l[u,v]=\frac{(|V(mcs(G^k_u,G^k_v))|+|E(mcs(G^k_u,G^k_v))|)^2}{(|V(G^k_u)|+|E(G^k_u)|)(|V(G^k_v)|+|E(G^k_v)|)}$
- If G^k_u and G^k_v are isomorphic and u matches v in the optimal matching of G^k_u and G^k_v, then Sl [u,v]= 1
- If G^k_u is subgraph isomorphic to G^k_v and u matches v in the optimal matching of G^k_v and G^k_u, we have:

$$Sl\,[u,\ v]=\frac{|V(G^k_u)|+|E(G^k_u)|}{|V(G^k_v)|+|E(G^k_v)|}$$

For Eq. 1, it is obvious that Sl [u,v]>0 holds, because both (n_{min}+1+D(u,v))2>0 and $(|V(G^k_u)|+|E(G^k_u)|)(|V(G^k_v)|+|E(G^k_v)|)>0$. Sl [u,v] = 1 can be showed as follows. Since min{d(u), d(v)}= d(u) and min{$d_{1,i}$, $d_{2,i}$} = $d_{1,i}$:

$$D[u,v]\le\frac{d(u)+\sum_{i=1}^{n_{min}}d_{1,i}}{2}=|E(G^k_u)|$$

Similarly, D(u,v) =|E(G^k_v)|. By combining such two inequations with the fact that n_{min}+1 = |V(G^k_u)| and n_{min}+1 = |V(G^k_v)|, we have Sl [u,v]= 1. For (2), since the node number of either G^k_u or G^k_v appearing inmcs can never exceed the minimum node number of G^k_u or G^k, |V(mcs(G^k_u, G^k))| = n min+1. Also, D(u,v) is known to be an upper bound of |E(mcs(G^k_u, G^k))|, which is proved in (Egozi *et al*., 2013). Thus, this inequation holds. Here, Sl [u,v] is an upper bound of such similarity, if we treat the right side of the equation in the property (2) as an accurate similarity of two nodes based on their MCS. For (3), this can be obtained based on the illustration of the first property, since when they are isomorphism, we have n_{min} + 1 =|V(G^k_u)| = |V(G^k_v)| and:

$$D(u,\ v)=\frac{d(u)+\sum_{i=1}^{n_{min}}d_{1,i}}{2}=|E(G^k_u)|$$

While leads to:

$$St(u, v) = \frac{|V(G_u^k)| + |E(G_u^k)|}{|V(G_v^k)| + |E(G_v^k)|}$$

Note that our local similarity is different from the vector-based node signature which deals with edge weights. For an undirected and unweighted graph, the edge weights for all its incident edges are 1. This means that the node signature in (Zhi-Yong *et al*., 2012) is merely its node degree and measuring the similarity of two nodes by their degrees is not sufficient, because there might be many pairs of nodes, which share the same degree but are with different structures. In our local similarity measure, we do not only consider the degrees of two nodes but also consider their k-neighborhoods.

Anchor selection and expansion: In our approach, we solve the two drawbacks as follows. Instead of matching all the nodes, we first match some important nodes as anchors. Every two anchors matched have high similarity and large degrees and can contribute a large number of matched edges. Then, we expand from the anchors to match the other nodes using the local similarity Sl as the measure. Thus, our solution consists of two steps, namely anchor selection and anchor expansion.

The anchors selected play two important roles in matching construction. (1) The matching anchors contribute a large number of edges to the matching M. (2) The anchors are the references to start with when matching the other nodes. For two nodes u □ V(G1) and v □ V(G2), we select (u,v) as matched anchors, if they satisfy the following two conditions.

- Min{d(u), d(v)} ≥ δ, where ä is the larger average degree of the two graphs, that is:

$$\delta = \max\left\{\frac{2\times|E(G_1)|}{|V(G_1)|}, \frac{2\times|E(G_2)|}{|V(G_2)|}\right\}$$

- S[u,v] ≥ τ, where τ is a threshold and generally τ>0.5 and is one sensitive threshold that has impacts on graph matching

The algorithm for anchor selection is shown in Algorithm 2. Given two graphs G1 and G2, it outputs a list of anchor pairs, denoted asA. In the algorithm,S1 and S2 denote the sets of matched nodes in V(G1) and V(G2), respectively.

Algorithm 2: Anchor-selection (G1, G2)

```
Require: two graphs G1 and G2;
Ensure: a list of matched anchor pairs A;
1: compute the similarity matrix S;
2: A ←□; S1 ←□; S2 ←□;
3: for all u □ V(G1) and v □ V(G2) in decreasing order of their similarity S[u,v] do
4: if S[u,v]≥τ and min{d(u), d(v)}≥δ and u / □ S1 and v/ □ S2 then
5: A ← A □{(u,v)}; S1 ← S1 □{u}; S2 ← S2 □{v};
6: return A;
```

Line 1 computes the similarity matrix S Eq. 2. Line 3 tries to match the pairs (u,v) for all u □ V(G1) and v □ V(G2) in the decreasing order of their similarity. In this way, the most similar pairs will have a large chance to be matched as the anchors. Line 4 selects the nodes that satisfy the two conditions for anchor selection that are not matched before. If the conditions in line 4 are all satisfied, we add the pair (u,v) into the list A and add the matched nodes u and v into S1 and S2, respectively in line 5. After checking all pairs, line 6 returns A as the anchor pairs.

We illustrate the anchor expansion algorithm (Algorithm 3) to obtain a matching M. Let A be the anchor pairs (u,v) selected already. Initially, M = A. Let N(u) and N(v) denote the immediate neighbors of u and v in graphs G1 and G2, respectively. For every matched pair (u,v) in the initial M, we put all (N(u)× N(v)) pairs in a queue Q, where Q is the set of candidate matching pairs sorted in decreasing order of their local similarity. In an iterative manner, we remove the pair (u,v) with the largest local similarity Sl [u,v] [Eq. (4)] from Q. If both u and v have not been matched before, we add (u,v) to M and put their all (N(u) × N(v)) immediate neighbor pairs into Q for further consideration. We repeat it until Q = □.

Algorithm 3: Anchor-expansion (G1, G2, A)

```
Require: two graphs, G1 and G2 and the anchor pairs A;
Ensure: a graph matching M;
1: M ← A; Q ←□; S1 ←□; S2 ←□;
2: for all (u,v) ← A do
3: S1 ← S1 □{u}; S2 ← S2 □{v}; Q ← Q □ (N(u)×N(v));
4: while Q □=□ do
5: remove (u,v) from Q with the largest similarity Sl [u,v];
6: if u/□ S1 and v/□ S2 then
7: M ← M□ {(u,v)}; S1 ← S1 □{u}; S2 ← S2 □{v}; Q ←Q □ (N(u)×N(v));
8: return M;
```

CONCLUSION

The time complexity of Algorithm 3 remains unchanged, compared to Algorithm 2, because it only repeats anchor-expansion constant times. It is worth noting that anchor-selection is the dominant factor and anchor-expansion can be done very quickly in practice compared to anchor-selection. Some results are shown that the time of anchor-expansion means the total expansion time including the τ selection.

REFERENCES

Bhattacharjee, A. and H. Jamil, 2012. WSM: A novel algorithm for subgraph matching in large weighted graphs. J. Intell. Inform. Syst., 38: 767-784.

Cheng, J., J.X. Yu and P.S. Yu, 2011. Graph pattern matching: A join/semijoin approach. IEEE Trans. Knowl. Data Eng., 23: 1006-1021.

Egozi, A., Y. Keller and H. Guterman, 2013. A probabilistic approach to spectral graph matching. IEEE Trans. Pattern Anal. Mach. Intell., 35: 18-27.

Kpodjedo, S., P. Galinier and G. Antoniol, 2010. On the use of similarity metrics for approximate graph matching. Electron. Notes Discrete Mathe., 36: 687-694.

Kpodjedo, S., P. Galinier and G. Antoniol, 2012. Using local similarity measures to efficiently address approximate graph matching. Discrete Applied Mathe.,. 10.1016/j.dam.2012.01.019

Plantenga, T., 2013. Inexact subgraph isomorphism in MapReduce. J. Parallel Distrib. Comput., 73: 164-175.

Tang, J., B. Jiang, C.C. Chang and B. Luo, 2012. Graph structure analysis based on complex network. Digital Signal Proc., 22: 713-725.

Zhi-Yong, L., Q. Hong and X. Lei, 2012. An extended path following algorithm for graph-matching problem. IEEE Trans. Pattern Anal. Mach. Intell., 34: 1451-1456.