Large Scale Graph Matching(LSGM): Techniques, Tools, Applications and Challenges

Azka Mahmood
CIIT
Sahiwal, Pakistan 57000

Hina Farooq
CIIT
Sahiwal, Pakistan 57000

Javed Ferzund
CIIT
Sahiwal, Pakistan 57000

***Abstract*—Large Scale Graph Matching (LSGM) is one of the fundamental problems in Graph theory and it has applications in many areas such as Computer Vision, Machine Learning, Pattern Recognition and Big Data Analytics (Data Science). Matching belongs to the combinatorial class of problems which refers to finding correspondence between the nodes of a graph or among set of graphs (subgraphs) either precisely or approximately. Precise Matching is also known as Exact Matching such as (sub)Graph Isomorphism and Approximate Matching is called Inexact Matching in which matching activity concerns with conceptual/semantic matching rather than focusing on structural details of graphs. In this article, a review of matching problem is presented i.e. Semantic Matching (conceptual), Syntactic Matching (structural) and Schematic Matching (Schema based). The aim is to present the current state of the art in Large Scale Graph Matching (LSGM), a systematic review of algorithms, tools and techniques along with the existing challenges of LSGM. Moreover, the potential application domains and related research activities are provided.**

Keywords*—*Big Data; Graph Matching; Graph Isomorphism; Graph Analytics; Data Models; Large Scale Graphs

I. Introduction

In this era of big data, graphs are considered as data representation tool that is capable for holding large scale attributed data and the relationships among data entities. It has been proven that graphs can represent structural information in the form of attributed objects (vertices) and their relationships (edges) in an efficient manner. The ubiquitous nature of graph structure provides better modeling approach for representation of relationships among almost anything (any kind of entities). Some examples from real world where graphs are playing an important role are: Social Networks[1], World Wide Web[2], Flight Route Graphs[3], Communication Networks[4] and Biological/ Chemical Networks[5] etc.

According to literature of graph processing, the problem sizes (benchmarked data sets) are getting large such as Social Network Graph has reached the limit of trillions of edges. Other examples are the Twitter graph which is one of the largest graphs that have 1.5 Billion edges and graph for Yahoo (The Altavista graph) contains 6.6 Billion edges [6], [7]. All the real world graphs with billions or trillions of vertices and edges are challenging to store, process and analyze.

There are various domains like Distributed Systems, Image Processing, Bio/Cheminformatics, Computer Vision and Pattern Matching in which characteristics of graphs are exploited . It is required in many applications to find similarity among objects/graphs. The problem of finding similarity among (sub) graphs is known as (sub) Graph Pattern Matching. Graph simulation[8], Graph isomorphism[9] and Attributed matching[10] are widely studied problems in graph matching. Isomorphism belongs to the NP-complete class of problems and is used for strictest matching of graphs which is conceptually applicable but could not scale well for large graphs[8]. On contrary, graph simulation is considered as an alternative to isomorphism with the relaxation in matching constraints and practically possible in polynomial time[11].

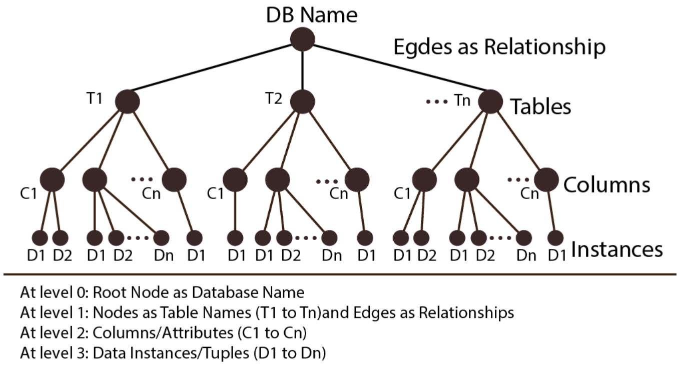
The outline followed in this paper is as: in section 2 it is discussed, how a data model can be represented as a graph model. Section 3 describes the graph matching problems grouped in three categories: semantic, syntactic and schematic matching. Further in section 4, graph matching measures are discussed. In section 5, a systematic review of existing algorithms, tools and techniques related to graph matching along with their potential applications is presented. In section 6, open challenges for both academia and industry are discussed. Related work is presented in section 7, followed by the conclusion in section 8.

II. Data Models as Graphs

The matching problem for graph-oriented data is challenging. Big data and IoT has made World Wide Web (WWW) a major source of data. In many diverse application domains, graphs are one of the important data structures to represent variety of data (Unstructured, Semi-structured, and Structured). Graphs are dominant among data models because of their expressive nature and power to model highly connected and attributed data[9]. Data models such as relational, object-oriented, XML, ontologies, RDF and hierarchies can be represented as graph-oriented data model. In this paper, data models are presented which can map structural data into graph data.

Relational data model is one of the basic and traditional data models which implements first order predicate logic for data management. Data entities have attributes and relationships. Key constraints (such as primary and foreign keys) or referential constraints are applied to attributes and possibly some attributes have data instances as well. The question arises about the data mapping from one model to another model. How can the relational data be represented as the graph data? How can nodes and edges of a graph represent entities, attributes, relationships, key-constraints and data instances of relational data? There are many possible ways for such kind of data mappings.

Generally, graphs represent data as nodes and edges. In the case of relational databases, database name becomes root node and schema is partitioned into tables at level-1 where edges between level-0 (Root Node) and level-1 (Table Nodes) represent relationships. At level-2, nodes can be specified as columns and edges can be considered as attributes. Further, leaf nodes specify data instances or can be referred as tuples. We present a general mapping tree of height 3 (see 1), considering the fact that tree is a specialized form of graph and it is possible to map data from one model to another.



At level 0: Root Node as Database Name
At level 1: Nodes as Table Names (T1 to Tn)and Edges as Relationships
At level 2: Columns/Attributes (C1 to Cn)
At level 3: Data Instances/Tuples (D1 to Dn)

Fig. 1. Data Mapping from Relational Data Model to Graph Data Model

XML data model is capable to model big data like it can capture features of data which is semi-structured and unstructured in nature. DAGs (Directed Acyclic Graphs) are used to represent XML data. Data instances in XML model can either be elements or attributes. Relationships among data entities can be referred as IS-A property. The mechanism of obtaining DAG from XML data is known as ID/IDREF. As XML data possess irregular structure, duplication, missing values and loose constraints, the mechanism of ID/IDREF causes removal of duplicated data and makes it sure that one object has one or more than one instances. The resultant data will be a graph with Parent-Child hierarchy (DAG). Therefore, matching problems related to data source can be resolved by mapping data from one model to another model. Similarly, conceptual hierarchies, ontologies, RDFs and object oriented data models can also be transformed into graph model[12]. The scope of this work is graph or subgraph matching problem. Data models (Relational model for structured data and XML model for semi-structured/Unstructured data) are discussed in which input data could be available for matching problems and it have to be transformed into graph data model.

III. Graph Matching

In Graph theory, Computer Scientists and Mathematicians have done variety of significant work. Graph Matching is one of the graph-based techniques which is briefly discussed in this paper. As Graph matching problems belong to the class of combinatorial problems so it can be practically (computationally) expensive. Graph algorithms which usually take labeled and attributed graphs as an input are good candidates for solving matching problems. From 1976 onward, there has been an increase of algorithms and techniques on graph matching. Representative example of matching algorithms is Ullmans matching algorithm[9]. Other tools and techniques are presented in section 5. In graph matching, categories can be made between matching techniques based on three classifications.

First, does the matching require on structural (topology) level between vertices of one graph or among transactions (set of graphs)? Matching of graph structure is often called as Exact matching or Precise matching or Syntactic Matching. Syntactic matching refers to the techniques in which input is interpreted as a function of structural information which follows formal definition of an algorithm. It was proposed by Bernstein and Cupid [13] system was used for its implementation. There are several graph matching approaches that work on structure-based matching[14], [15]. Conceptual similarity matching is usually insufficient in syntactic matching.

The second category in matching techniques for graphs is based on finding conceptual correspondence between graphs and it is also called semantic matching or approximate matching. Semantic matching refers to the techniques in which input is interpreted as model-theoretic/formal semantics and valid justifications of results are provided[8].

The final distinction in graph matching can be made by matching graphs on the basis of schema and it is also referred as schema-based matching or Schematic Matching. Graph mining is another interesting problem that exploits the similar concepts as graph matching. Graph mining[16] is also known as structural motif finding and it is aimed to find common and interesting patterns in single graph or in transactions (set of small graphs)[17], [4]. In this work, different aspects of graph matching problems and techniques are focused. Several aspects of Graph mining are discussed in surveys which provides a thorough understanding about this technique [18], [19].

IV. Graph Matching Measures

Graph matching measures are well-known concepts that are also known as graph similarity measures such as Graph edit distance, Median Graph, Maximum Common Subgraph, Minimum Common SuperGraph, Graph Isomorphism and Subgraph Isomorphism. Graph isomorphism is used to check whether graphs have structural similarity or not. Subgraph Isomorphism is used to find if a graph is a subgraph(part of a graph) of another.

A. ***Graph Edit Distance(GED):***

The GED (Graph Edit Distance) of data graphs is determined by the number of edit operations needed to transform one graph into another. Thus, smaller the edit distance of two graphs, the more similarity will exist between graphs. Among the graph similarity measures, GED (Graph Edit Distance) is most commonly used to find similarities between pair of graphs or subgraphs.It is also referred as error-tolerant graph isomorphism. Many applications need to manipulate graph-oriented data with less distortion in order to transform a graph into a similar structured graph[20]. Like SED (String Edit distance), GED performs a set of basic graph edit operations for manipulation such as vertex/edge insertion, vertex/edge deletion and vertex/edge substitution. It has numerous applications in the fields of Computer Vision, Pattern Recognition, Machine Learning, Handwriting Recognition and Cheminformatics.

B. ***Median Graph:***

The concept of median graph is used for representation of graphs. It can be exploited to extract significant structural patterns from graphs on the basis of similarity or dissimilarity. The resultant graph that contains important information can be obtained by the procedure of graph matching[21]. Graphs for big data are usually stored in clusters. Median graph is a potential candidate for better partitioning of graphs in clustered environment. Web Mining, Shape Matching and Image Retrieval are some of the dominant applications of median graph.

C. *Maximum Common Subgraph(MCS)*

The similarity of the objects can be measured by Maximum Common Subgraph (MCS) if no isomorphism exists between graphs. Graph matching can be accomplished by MCS between a set of (sub) graphs[22]. For example in Cheminformatics and Bioinformatics, MCS plays an important role in several aspects like Molecular Spectra Interpretation, Biochemical Activity Prediction, Reaction Modeling and many others. There are many research fields other than Chem/Bioinformatics as well, in which MCS is playing a significant role such as Image Recognition, Computer Vision, Mathematics and Pattern Matching etc.The minimum common supergraph of corresponding graphs can be thought of as either the concept of smallest super trees between corresponding trees or the shortest common supersequence among collection of strings.

V. Potential Application Domains, Algorithms, Tools and Techniques for LSGM

There are many emerging applications where semantic matching is applicable such as schema emergence, event processing, data migration/integration, management of knowledge diversity, query translation and resource discovery etc[41].

In Health Sciences, Subgraph Matching has applications for example biological graphs [42], alignment of metabolic and protein networks [43], [44], inference of brain connectivity on MRI(Magnetic Resonance Imaging) data[45] and biological network analysis[5]. Network monitoring[2], schema matching [46], information fusion and control[47], product assembling and design [48], process management[49], document classification [50] and anomaly detection[51] are among real use cases of Graph Matching.

Schema based matching is used to operate traditional applications such as data warehousing , information integration and distributed query processing as well as emergent applications like service integration on WWW, peer to peer database management and agent communication etc. Generally, such applications exploit structural data or typically conceptual models.

Computer Vision and Image processing have various potential applications where GM is intensively used for example, similar image detection[52], Person or Object identification and retrieval[53] , 3D perspective reconstruction[54], image extrapolation[55], satellite imagery [56] etc.

Table I presents the systematic review of surveyed Graph Matching tools, techniques and algorithms. DualIso[40] is an algorithm that performs exact matching for subgraph isomorphism problem by exploiting pruning algorithm which makes it conceptually simple and memory efficient. TuboIso[39] is another robust and efficient solution for Isomorphic subgraph search for large scale graphs which is exploiting two novel concepts: COMP/PREM (Combine and Permute strategy) and the Candidate Region Exploration. STwig[12] is the first system that can perform online graph matching and exploration on large scale graphs. For graph exploration, graph data is deployed on memory cloud which exploits commodity hardware (machines) for clusters. Spidermine[38] is suitable to obtain top-K results for graph pattern matching from input graphs. SIGMA (A SET-COVER-BASED INEXACT GRAPH MATCHING ALGORITHM)[37] provides efficient inexact graph matching featured by filtering algorithms.

Catalog integration is one of the prominent applications of web service integration. Conceptual hierarchies can be represented as trees with attributed nodes and edges. In applications like catalog integration, catalogs/service dictionaries can be represented as conceptual hierarchies. eBay and Amazon catalogs are typical examples of such catalogs to quote. Catalog Matching Problem and mapping techniques are discussed in [57]. Web services are applications that provide web interface for users so that they can interact and utilize web services[58]. On the other hand, semantic web exploits the concepts of ontologies and knowledge representation in order to deliver better services. In [59], the process of integration and discovery of web services is presented.

VI. Existing Challenges for Large Scale Graph Matching

With the advent of cutting edge technologies, data is getting huge and traditional approaches are not sufficient to grasp the meaningful hidden insights from data. There are many significant open challenges that are posed by the scale of graphs ranging from storage infrastructures to processing paradigms. General purpose systems for graph processing are not available yet because it is critical to develop such platforms. Graph analytics are of two types that is real-time/online graph analytics and batch/offline graph processing. Distributed systems are considered as feasible platform for both kind of processing for graphs that contain hundreds of billions of nodes. Dataset of graph can be a single large graph or a set of small graphs that are often known as transactions.

Graph partitioning belongs to the class of problems which are NP-complete. Parallel and distributed tools and techniques are upsurging to analyze graph data efficiently. Since the graph of big data is difficult to process at once so it has to be partitioned into set of small graphs while preserving the connectedness of graph and balancing the load on clusters. Therefore, in distributed environments it is challenging to apply state-of-the-art graph algorithms and analytics techniques. A well discussed problem of graph processing is to find good and balanced partitions of large graphs so they could evenly distribute the computational load across clusters[60]. The prominent methods for graph partitioning are edge-cut partitioning and vertex-cut partitioning. Communication overheads can be reduced by exploiting edge-cut partitioning also it can balance the number of nodes (vertices) for each partition [1]. On contrary, vertex-cut partitioning technique can be used to partition Power-Law graphs that contain real-world data such as Collaboration Network or Social Networks.

Graph comparison can be performed by matching graphs or subgraphs. Generally, graph matching problem is to find

TABLE I. ALGORITHMS, TOOLS AND TECHNIQUES FOR LSGM

Algorithms,tools		Indexing	Matching	Locality	Spatial	Temporal
and Techniques		Scheme	Scheme		Complexity	Complexity
Ullman[15]	1976	No-Indexing	Exact	In-Memory	$O(N^3)$	$O(N!N^2)$
SubDue[23]	1994	Frequent Subgraph	Inexact	Out-of -core	-	Exponential
GraphGrep[24]	2002	Frequent Subgraph	Both	In-Memory	$O(l_p.n_p.l_i)^{l_p}$	$O(n_q + e_q)$
gIndex[25]	2004	Inverted Indexing	Both	Out-of-core	-	$\theta(\frac{1}{max^l})^2$
CTree[26]	2006	Frequent Subgraph	Both	In-Memory	$O(n^2)$	$O(n.d^2logn)$
FG-Index[27]	2007	Edge indexing	Exact	Out-of-core	$O(F_d)$	$O(T)$
Tree+Delta[28]	2007	Tree-based indexing	Exact	In-Memory	$O(C_q)$	Ploynomial
TALE[29]	2008	NH-Indexing	Inexact	Out-of-core	$O(S_{bit}.log(d))$	$O(S_{bit})$
GraphQL[30]	2008	Frequent Subgraph	Both	In-Memory	$O(k.n)$	$O(n^k)$
RJOIN[31]	2008	Frequent Subgraph	Both	In-Memory	$O(n.5^n)$	$O(mn5^n)$
QuickSi[32]	2008	Swift Indexing	Exact	In-Memory	$O(SEQ_q + g)$	Polynomial
GADDI[33]	2009	NH-Indexing	Exact	In-Memory	$O(n^2)$	$O(n^3)$
DistanceJoin[34]	2009	Frequent Subgraph	Inexact	In-Memory	$O(n)$	$O(n)$
BitMat[35]	2010	Edge indexing	Exact	In-Memory	$O(m)$	$O(m)$
Rdf-3x[36]	2010	Edge indexing	Inexact	Out-of-core	$O(m)$	$O(m)$
SIGMA[37]	2010	Frequent Subgraph	Inexact	On-cloud	-	-
SpiderMine[38]	2011	Frequent Subgraph	Exact	Out-of-core	-	Exponential
STwig[12]	2012	No-Indexing	Both	On-Cloud	$O(n)$	$O(n)$
TurboIso[39]	2013	No-Indexing	Exact	In-Memory	Polynomial	$O(G^2)$
DualIso[40]	2014	Vertex Indexing	Exact	In-Memory	$O(E)$	$O(E)$

the similarity (or dissimilarity) between model graph and input graph. Graph matching can be exact that is exploiting graphs by using their syntactical description or inexact which means comparison between graphs can be performed on the basis of semantics of graphs. There are different approaches used to achieve exact matching for example Isomorphism, monomorphism and subgraph isomorphism.

During last few decades, it has been an open challenge to design well suited algorithms with low complexity for matching large scale graphs. There are many different invariant of graph properties such as scaling and rotation etc. Usually, a good structural correspondence between graphs can be achieved by Graph Isomorphism [61]. Graph isomorphism has applications like Bio/Cheminformatics, Automation of Electronic Circuits and Exact Pattern Recognition[62].

VII. RELATED WORK

In case of big data, graph-oriented data is too large to be queried in an easy and efficient manner. Graph pattern matching queries can be large,take exponential time and return number of matches from a graph[63]. Among classes of queries, TWIG queries are grabbing attention of users from both academia and industry. Many important queries like RDF queries and XQuery/ XPath queries which are XML queries can be treated as TWIG queries[64].

Experiments can be performed on real datasets as well as on synthetic datasets. Examples of existing real datasets are US Patents and WordNet which represents relationships between US referenced patents and English words, respectively. First mentioned graph contains 133,455 edges and 82,670 vertices [65]while second graph has 16,533,438,8 edges and 3,774,768 vertices[66]. Additionally, R-MAT[67] can be used to generate synthetic datasets. The problem of matching graphs for big data is challenging due to the concern of size of graph. Some existing approaches for matching graphs are: Indices method[68] and pruning methods [69]. Distributed and parallel approaches are also used for processing large graphs for example MST (Minimal spanning tree), SPP (Shortest path problem) and connected component are algorithmic strategies that can be used for computation[3].

Matching operations can be classified in various dimensions such as on the basis of input/output of algorithms or on the basis of characteristics of an entire matching process[10]. As first dimension, data/conceptual model representation (either schema or ontology) can be considered as an input for algorithm. For example OWL and RDF models are supported by QOM[70], Relational and XML models are supported by Cupid[13] and Object Oriented and Relational models are supported by Artemis[63]. As a second possibility, algorithms can exploit the characteristic of data (what is the kind of input data?) that is either the input provides instance level information, the schema level information or both. For example COMA[71]and Cupid[13] rely on schema level information, GLUE[36] rely on Instance level information while QOM[70] rely on both. In matching process, algorithms can be classified as semantic, syntactic and schematic nature of computation. Algorithms are used analyze the patterns of data exactly or approximately.

Recently proposed Matching Systems based on schema level matching are Artemis (Analysis of Requirements: Tool Environment for Multiple Information Systems), Similarity Flooding (SF), QOM (Quick Ontology Mapping), OLA (OWL Lite Aligner), COMA (Combination of Matching Algorithms), Cupid, NOM (Naive Ontology Mapping) and Anchor Prompt [72], [70], [71], [13]. There also exist some infrastructures that use matching as integration such as Protoplasm, OntoMerge, MAFRA and Chimaera [73].

VIII. CONCLUSION

We are living in the age of Big Data and graphs are the most suitable choice for representing large scale multi-modal data as they can effectively represent the relationships of different data. Large scale graphs have been used for analysis of complex data sets like social networks, bioinformatics, health informatics, social security, web and scientific applications that produce large amount of data. To fully utilize the information represented by graphs, efficient matching algorithms, tools and techniques are required. In this paper, a review of state of the art Large Scale Graph Matching (LSGM) algorithms, techniques and tools has been presented. Matching problem is described according to various types of graph matching.

Moreover, potential applications and research activities has been presented. This article will be helpful for the researchers to get firsthand knowledge of existing LSGM algorithms and techniques and to plan for future research.

Acknowledgment

The authors would like to thank...

References

[1] Amine Abou-Rjeili and George Karypis. Multilevel algorithms for partitioning power-law graphs. In *Proceedings 20th IEEE International Parallel and Distributed Processing Symposium*, 2006.

[2] M. Kraetzl P. Showbridge and D. Ray. Detection of abnormal change in dynamic networks. In *Information, Decision and Control*, 1999.

[3] T.Y. Cheung. State of the art of graph-based data mining. *IEEE Transactions on Software Engineering*, 1983.

[4] Michihiro Kuramochi and George Karypis. Finding frequent patterns in a large sparse graph. *Data mining and knowledge discovery*, 2005.

[5] F. R. Bach M. Zaslavskiy and J.-P. Vert. Global alignment of protein lcprotein interaction networks by graph matching methods. *Bioinformatics*, 2009.

[6] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks. In *Proceedings of the 20th international conference on World Wide Web*, 2011.

[7] Lab for web algorithms. Datasets. urlhttp://law.di.unimi.it/datasets.php, 2016.

[8] B. Gallagher. Matching structure and semantics. *The Association for the Advancement of Artificial Intelligence*, 2006.

[9] Foggia P. Sansone C. Vento M. Cordella, L. P. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence*, 2004.

[10] L. Chen Y. Yuan, G. Wang and H. Wang. Efficient subgraph similarity search on large probabilistic graph databases. *Proceedings of the VLDB Endowment (PVLDB)*, 2012.

[11] M. R. Garey and D. S. Johnson. *A Guide to the Theory of NP-Completeness*, chapter Computers and Intractability. 1990.

[12] Wang H. Wang H Sun, Z. Efficient subgraph matching on billion node graphs. In *Proceedings of the VLDB Endowment*, 2012.

[13] Bernstein P Madhavan, J and E Rahm. Generic schema matching with cupid. In *Proceedings of very large scale databases(VLDB)*, 2001.

[14] Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern recognition letters*, 1998.

[15] Julian R. Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 1976.

[16] Eveline Hoekx Goethals, Bart and Jan Van den Bussche. Mining tree queries in a graph. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005.

[17] Diane J. Cook and Lawrence B. Holder. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1994.

[18] Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. *ACM Sigkdd Explorations Newsletter*, 2003.

[19] Takashi Washio and Hiroshi Motoda. A quantitative comparison of the subgraph miners mofa, gspan, ffsm, and gaston. In *European Conference on Principles of Data Mining and Knowledge Discovery*, 2005.

[20] Dacheng Tao Xuelong Li Xinbo Gao, Bing Xiao. A survey of graph edit distance. *Pattern Analysis and Applications*, 2010.

[21] Horst Bunke. Graph matching: Theoretical foundations, algorithms and applications. In *Proceedings of Vision*, 2000.

[22] Carsten Maple Yu Wang. A novel efficient algorithm for determining maximum common subgraphs. In *Proceedings of Ninth International Conference on Information Visualisation*, 2005.

[23] D. J. Cook L. B. Holder and S. Djoko. Substucture discovery in the subdue system. In *KDD Workshop*, 1994.

[24] Rosalba Giugno and Dennis Shasha. Graphgrep: A fast and universal method for querying graphs. In *Proceedings of 16th International Conference on Pattern Recognition*, 2002.

[25] P. S. Yu X. Yan and J. Han. Graph indexing: A frequent structure-based approach. In *Association for Computing Machinery's Special Interest Group on Management of Data(SIGMOD).ACM*, 2004.

[26] H. He and A. K. Singh. Closure-tree: An index structure for graph queries. In *IEEE International Conference on Data Engineering (ICDE)*, 2006.

[27] W. Ng J. Cheng, Y. Ke and A. Lu. Fg-index: towards verification-free query processing on graph databases. In *Association for Computing Machinery's Special Interest Group on Management of Data(SIGMOD).ACM*, 2007.

[28] Jeffrey Xu Yu Zhao, Peixiang and Philip S. Yu. Graph indexing: tree+ delta¡= graph. In *Proceedings of the 33rd international conference on Very large data bases. VLDB Endowment*, 2007.

[29] Y. Tian and J. M. Patel. Tale: A tool for approximate large graph matching. In *IEEE International Conference on Data Engineering (ICDE)*, 2008.

[30] H. He and A. K. Singh. Graphs-at-a-time: query language and access methods for graph databases. In *Association for Computing Machinery's Special Interest Group on Management of Data(SIGMOD).ACM*, 2008.

[31] B. Ding P. S. Yu J. Cheng, J. X. Yu and H. Wang. Fast graph pattern matching. In *IEEE International Conference on Data Engineering (ICDE)*, 2008.

[32] Xuemin Lin Haichuan Shang, Ying Zhang and Jeffrey Xu Yu. Taming verification hardness: An efficient algorithm for testing subgraph isomorphism. In *Proceedings of VLDB Endow*, 2008.

[33] Shirong Li Zhang, Shijie and Jiong Yang. Gaddi: distance index based subgraph matching in biological networks. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. ACM*, 2009.

[34] L. Chen L. Zou and M. T. Ozsu. Distancejoin: Pattern match query in a large graph database. In *Proceedings of the 33rd international conference on Very large data bases. VLDB Endowment*, 2009.

[35] Chaoji V. Zaki M. J. Hendler J. A. Atre, M. Matrix bit loaded: a scalable lightweight join query processor for rdf data. In *Proceedings of the 19th international conference on World wide web*, 2010.

[36] T. Neumann and G. Weikum. The rdf-3x engine for scalable management of rdf data. *VLDB Journal*, 2010.

[37] R. Giugno A. Pulvirenti A. Ferro M. Mongiovi, R. D. Natale and R. Sharan. Sigma: a set-cover-based inexact graph matching algorithm. *J. Bioinformatics and Computational Biology*, 2010.

[38] D. Lo X. Yan J. Han F. Zhu, Q. Qu and P. S. Yu. Mining top-k large structural patterns in a massive network. *Proceedings of the VLDB Endowment*, 2011.

[39] Jinsoo Lee Wook-Shin Han and Jeong-Hoon Lee. Turboiso: Towards ultrafast and robust subgraph isomorphism search in large graph databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2013.

[40] Xuemin Lin Haichuan Shang, Ying Zhang, Abhishek Kothari Arash Fard John A Miller Jeffrey Xu YuMatthew Saltz, Ayushi Jain, and Lakshmish Ramaswamy. Dualiso: An algorithm for subgraph pattern matching on very largelabeled graphs. In *IEEE International Congress on Big Data (BigData Congress)*, 2014.

[41] Seth Greenblatt Coffman, Thayne and Sherry Marcus. Graph-based technologies for intelligence analysis. *Communications of the ACM*, 2004.

[42] C. Santos Y. Tian, R. Mceachin and J. Patel. Saga: a subgraph matching tool for biological graphs. *Bioinformatics*, 2007.

[43] Q. Yang and S. Sze. Path matching and graph matching in biological networks. *Journal Of Computational Biology*, 2007.

[44] M. Baym R. Singh C. S. Liao, K. Lu and B. Berger. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 2009.

[45] J. Vogelstein P. Muse M. Fiori, P. Sprechmann and G. Sapiro. Robust multimodal graph matching: Sparse coding meets graph matching. In *Proceedings of NIPS*, 2013.

[46] H. Garcia-Molina S. Melnik and E. Rahm. A versatile graph matching algorithm and its application to schema matching. *Data Engineering*, 2002.

[47] R. C. Wilson M. L. Williams and E. Hancock. Multiple graph matching with bayesian inference. *Pattern Recognition Letters*, 1997.

[48] R. Nagi C. Romanowski and M. Sudit. Data mining in an engineering design environment: Or applications from graph matching. *Computers and operations research*, 2006.

[49] Z. Gao R. Dijkman, M. Dumas and L. Garcia-Banuelos. Graph matching algorithms for business process model similarity search. *Business Process Management, Springer*, 2009.

[50] H. Bunke A. Schenker, M. Last and A. Kandel. Classification of web documents using graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004.

[51] A. Anwar and A. N. Mahmood. Anomaly detection in electric network database of smart grid: Graph matching approach. *Electric Power Systems Research*, 2016.

[52] H.K. Tan W.L. Zhao, C.W. Ngo and X. Wu. Near-duplicate keyframe identification with interest point matching and pattern learning. *IEEE Transactions on Multimedia*, 2007.

[53] J. Yan M. Xu J. Wu Y. Shen, W. Lin and J. Wang. Person re-identification with correspondence structure learning. In *ICCV*, 2015.

[54] M. Zhu X. Zhou and K. Daniilidis. Multi-image matching via fast alternating minimization. In *ICCV*, 2015.

[55] Y. Liang R. R. Martin M. Wang, Y. Lai and S.M. Hu. Biggerpicture: data-driven image extrapolation using graph matching. *ACM Transactions on Graphics*, 2014.

[56] C. Zhou A. Kitamoto and M. Takagi. Similarity retrieval of noaa satellite imagery by graph matching. *Electronic Imaging: Science and Technology*, 1993.

[57] L. Serafini P. Bouquet and S. Zanobini. Semantic coordination: A new approach and an application. In *Proceedings of the International Semantic Web Conference (ISWC)*, 2003.

[58] R. J. Miller Y. Velegrakis and J. Mylopoulos. Representing and querying data transformations. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2005.

[59] T. Payne M. Paolucci, T. Kawmura and K. Sycara. Semantic matching of web services capabilities. In *Proceedings of the International Semantic Web Conference (ISWC)*, 2002.

[60] Bruce Hendrickson and Robert W. Leland. A multi-level algorithm for partitioning graphs. *Science*, 1995.

[61] Vento P. Foggia, C. Sansone. An improved algorithm for matching large graphs. In *Proceedings of the 3rd IAPR-TC15 International Workshop on Graph-based Representation*, 2001.

[62] M. Vento P. Foggia, C. Sansone. A performance comparison of five algorithms for graph isomorphism. In *Proceedings of the 3rd IAPRTC15 Workshop on Graph based Representation (GbR2001)*, 2001.

[63] Philip A. Bernstein. Query processing in a system for distributed databases (sdd-1). *ACM Transactions on Database Systems (TODS)*, 1981.

[64] Nick Koudas Bruno, Nicolas and Divesh Srivastava. Holistic twig joins: optimal xml pattern matching. *Proceedings of ACM SIGMOD international conference on Management of data*, 2002.

[65] Pajek Datasets. US-Patents Dataset. urlhttp://vlado.fmf.uni-lj.si/pub/networks/data/patents/Patents.htm, 2004.

[66] Vladimir Batagelj. English Words. urlhttp://vlado.fmf.uni-lj.si/pub/networks/data/dic/Wordnet/Wordnet.zip, 2007.

[67] Yiping Zhan Chakrabarti, Deepayan and Christos Faloutsos. R-mat: A recursive model for graph mining. In *Proceedings of SDM*, 2004.

[68] P. S. Yu H. Jiang, H. Wang and S. Zhou. Gstring: a novel approach for efficient search in graph databases. In *Proceeding of International Conference on Data Engineering (ICDE)*, 2007.

[69] C. Sansone L. P. Cordella, P. Foggia and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[70] M. Ehrig and S. Staab. Qom: Quick ontology mapping. In *Proceedings of the International Semantic Web Conference (ISWC)*, 2004.

[71] H. H. Do and E. Rahm. Coma - a system for flexible combination of schema matching approaches. In *Proceedings of the Very Large Data Bases Conference (VLDB)*, 2001.

[72] H. Garcia-Molina S. Melnik and E. Rahm. Similarity flooding: A versatile graph matching algorithm. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2002.

[73] N. Silva A. Maedche, B. Motik and R. Volz. Mafra - a mapping framework for distributed ontologies. In *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, 2002.