Proceedings of the 10th INDIACom; INDIACom-2016; IEEE Conference ID: 37465
2016 3rd International Conference on "Computing for Sustainable Global Development", 16th - 18th March, 2016
Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA)

Graph Pattern Matching: A Brief Survey of Challenges and Research Directions

Komal Singh
National Institute of Technology, Kurukshetra,
Haryana, INDIA
Email ID: komalsingh.0492@gmail.com

Vikram Singh
National Institute of Technology, Kurukshetra,
Haryana, INDIA
Email ID: viks@nitkkr.ac.in

Abstract* – Due to tremendous intensification of Internet, most of the data is archived and analyzed in graph-structured database. Graph Database is a collection of operational data objects, mapped into huge labeled graph or a set of labeled directed graphs. In recent years, modeling data in graph structure becomes evident and effective for processing in some of the prominent application area like social analytics etc. Graph databases are mapped into data graphs for further computing or processing. Graphs are one of the dominant data modeling tool used in many application areas. Hence, there is a demand for proficient querying techniques on such large data graphs. The user query is mapped into pattern graph, which is constructed by connecting nodes based on links/relationships required by user. The primary objective of graph pattern matching (GPM) is to determine all the candidates matching to a pattern query in a large data graph. This survey paper discusses various graph pattern matching approaches, used by researchers in recent year for query processing. Further, we also highlight the various challenges imposed by modern day computing and possible future research directions in query processing using graph pattern matching.

***Keywords* – Big Data Analytics; Graph Data; Graph Pattern Matching; Graph Mining; Query Processing**

I. INTRODUCTION

Graphs and graph matching algorithms serve as a powerful tool in the process of search and comparison due to their efficiency and easy utility in form of representation. Some of the significant research and application areas of graph pattern mining include information retrieval, knowledge discovery and data mining, mathematical graph theory, artificial intelligence, computer vision, computer aided design etc. The research efforts made into pattern matching in the past few years go beyond its application in computer science, as it also spans in various other research communities [1]. A graph is an abstract representation of a set of objects (vertices/nodes) where some pairs are connected by links (edges/relationships). A graph is a powerful tool for modeling database objects and their relationships among data items in various application domains. A database with an explicit graph structure is known as graph database.

A data graph is a collection of data items stored into graph database. Data graph stores data, structured in the nodes and relationships of a graph. Nodes represent the actual data and Relationships represents the connection among these nodes. For data of any momentous volume or value, graph database is best suited way of modeling. The prime challenge in graph DB is to understand the semantic relationship among its ingredient DB elements. In the initialization of a graph database index-free adjacency are preferred, in which connected nodes are physically linked to each other [12]. There are real life graphs everywhere representing complex objects and networks, e.g., social networks, Internet, blog networks, food web, protein interaction networks. In most domain applications, one want to find the correspondence between two objects represented in data graph via different graph models. In graph database models, data structures for the database schema and instances are modeled in the form of graphs or their generalizations [1]. These graph models try to trounce the inherent limitations in traditional db models, as interconnectivity among data items is a vital facet. Although there are many graph models, we consider the following representation [1]:

A node-labeled, directed graph is defined as $G = (V, E, L)$, where (a) V is a set of nodes; (b) $E \subseteq V \times V$ is a set of edges, in which (v,v') denotes an edge from node v to v'; and (c) for each v in V, $L(v)$ is the label of v. The label $L(v)$ may indicate a variety of real life semantics, a list of attributes, or even a set of predicates.

Development of graph models provides an opportunity to model related problems in high performance computing. Graph matching problem becomes one of the central problems in a variety of emerging application domain in computer science, biology, chemistry and many other academic areas. A graph pattern matching (GPM) approach is to discover candidate matches in a data graph for a query graph (pattern). Graph matching is a complex problem, as various level of graph modeling is involved. Query processing is one of the core activities in data processing and analytics, thus this paper covered the recent research efforts on graph pattern matching techniques for query processing in various applications. The fundamental work on this direction is incorporated. The primary objective of this survey is to illustrate the existing research effort and progress that has been conducted in the area of graph pattern matching, concentrating on revised graph pattern matching techniques.

A. Preliminaries

Fundamental notations related to pattern graph and matching are discussed below,

- **Data Graph**: A directed graph $G = (V, E, f_A)$, with V finite set of nodes, $E \subseteq V \times V$ edge from node v to v' and $f_A(\cdot)$ a function that relate each node v in V with a tuple $f_A(v) = (A_1=a_1, \ldots, A_n = a_n)$, a_i is a constant, and A_i an attribute of v, and carrying the node contents, such as label, keywords, blogs.
- **Pattern Graph**: A directed graph $P= (V_P, E_P, f_v, f_e)$, with V_P and E_P are pattern nodes and the set of pattern edges defined for data graphs; $f_v(\cdot)$ is a function defined on V_P such that for each node $u \in V_P$, $f_v(u)$ is a predicate of u and $f_e(\cdot)$ is a function defined on E_P.
- **Matching:** The primary objective of a graph pattern matching problem is to find the maximum match in data graph G for pattern graph P, denoted by $M(P,G)$.
- **Maximally Contained graph:** Consider a pattern query $P_S = (V_P, E_P, f_v)$, and a view set $\mathcal{V} = \{V_1, . V_n\}$, where $V_i = (V_i, E_i, f_i)$. P_S is contained in $\mathcal{V}$, denoted by $P_S \sqsubseteq \mathcal{V}$,

  if there exists a mapping λ from E_P to powerset $\mathcal{P}(\cup_{i\in[1,n]} E_i)$, such that for all data graphs G, the match set $S_e \subseteq \cup_{e' \in \lambda(e)} S_{e'}$ for all edges $e \in E_P$.

- **Paths:** The sequence of nodes $(v_1, \ldots v_n)$ such that (v_i, v_{i+1}) is an edge in G for each $i \in [1,n-1]$.
- **Distance:** The length of the shortest undirected path from u to v in G, denoted as denoted by $dist(u, v)$.
- **Diameter:** The diameter of a graph G, denoted by d_G, shortest distance between all pairs of nodes in G, *i.e.*, $d_G = max(dis(u, v))$ for all nodes u, v in G.

B. Contribution and Outline

There are two major contribution of this paper; primarily to highlight the various modern day computing challenges faced by graph pattern matching in query processing. This also list the challenges imposed by evolution of big data. Another contribution is the overall comparative performance analysis of various graph pattern matching approaches used for query processing in recent years by various researchers. Further we have underlined the various possible future scopes in the pattern matching.

In section 2, classification of graph pattern matching is illustrated and subsequently the detailed description is given in sub sections. The performance analyses on various graph pattern matching approaches are presented along with algorithmic complexity in section 3. Section 4 highlights the current challenges for the existing approach and the future possible direction for the research is also highlighted.

II. GRAPGH PATTERN MATCHING

There is diversity of problems that build on graph matching, as it is a complex processing task. One such activity is that of graph mining or structural match finding. Although the objective of graph pattern matching is to determine the instances or occurrences of an explicit pattern in a query pattern, whereas the primary objective of graph mining is to determine a set of most common or most "interesting" patterns in a graph. There are number of approaches for graph pattern matching problem. All the approaches are categorizes based on graph structure and semantics, shown in figure 1. The several graph pattern matching techniques are developed for finding the pattern matches, despite the complex and variety of graph properties [2].

Graph pattern matching applications include finding research collaboration information like citation links analysis from bibliographic data, relationships and their proximity between persons in a social network and finding patterns of interest by scientists in biological networks like protein-protein interaction networks, source code analysis etc. Extensive research has been done and many algorithms are proposed and implemented in following Graph pattern mining areas: frequent subgraph mining using sub-graph isomorphism, exact pattern matching and pattern matching with wildcards based on distance [2]. Graph pattern matching is widely used to search and analyze, e.g., social graphs, biological data and transportation networks [14]. Given a data graph 'G' pattern query 'P', graph pattern matching technique objective is to find all the candidate matches in a data graph G for user query pattern query P.

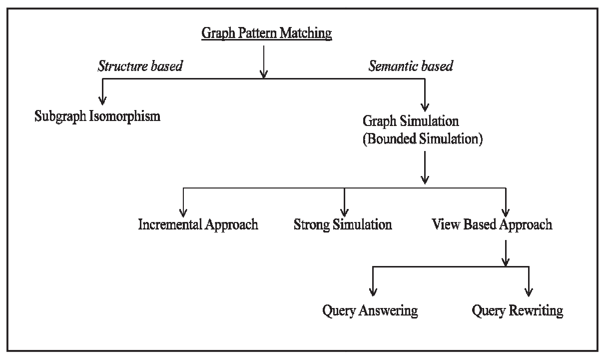

Fig. 1. Classification of Graph Pattern Matching techniques (based on Semantic and Structure)

A. Subgraph Isomorphism

All graphs contribute to the basic structure elements, vertices and edges. This class contains approaches that find matches based to structure similarity i.e. focus on the specific structure property, e.g. restriction on the number of edges, number of isomorphs etc. The large data graph leads to huge number of isomorphs, thus calculation of isomorphs is complex and expensive [7]. Subgraph isomorphism approach either finds all the sub graphs in a relatively large data graph G that is isomorphic to the given pattern graphs P, or returns a subgraph isomorphic to the pattern with the best matched nodes, based on various quality models. The pattern matching problems based on subgraph isomorphism are essentially NP-complete [13]. Subgraph isomorphism is often used in various application domains such as chemical or bioinformatics, in which candidate match with respect to an identical structure or design is chosen. However, the complexity of isomorphism based approach infeasible to determine the candidate matches in “big” real life data graphs such as social graphs. In fact, for prominent area like health care analytics, scientific analytics Web and social networks analyses, these techniques suffer in finding “inexact” matches [7]. As in these scenarios resultant subgraphs do not have identical structures to the query patterns.

B. Graph Simulation

Graph simulation is to define a binary function on data graph that will find a match for pattern graph on data graph. Graph simulation can be determined in quadratic time [11].In many application areas, it is desired to examine the connectivity among the pair of nodes via an arbitrary path length or with a constrained on number of hops. Edge-to-edge mapping in subgraph isomorphism and graph simulation are not able to specify such connectivity in a data graph because they impose topological constraint too strictly. For these reasons, computation methods used for the concepts of graph pattern matching are revised to accurately and efficiently identify sensible matches in real-life graphs. Semantic based graph pattern matching methods are such methods which are capable to overcome the underline issues in structure based matching methods, discussed in next section.

C. Semantic based Graph Pattern Matching

Semantic matching approaches attempt to find semantically/conceptually similar graphs, i.e., candidate matches in graphs are based on semantic of nodes and edge. This class of pattern matching techniques improves the traditional notion for graph pattern matching by determining the maximum bounded simulation relation rather than a function and also validates edge to path mapping between pattern and data graph of various bounds [11]. Edge-to-path mapping of various bound helps a semantic based graph pattern matching technique by reducing the overall matching complexity and illustrates patterns in emerging applications.

*1) Bounded Simulation:*Bounded simulation is one way of redefining the graph pattern matching from the traditional concept of pattern matching [4]. Traditional approaches prefer edge-to-edge mapping in the process of finding the candidate pattern, while bounded simulation allows edge-to-path mapping. This overcomes the shortcoming of label equality in pattern matching. In the pattern graph, search condition is specified on the node and applied on the content of the edge. Bounded simulation imposes a weaker topological constraint [4] on the evaluation of the relation for the maximum bounded simulation rather than functions. Bounded simulation is an enhanced or modified version of graph simulation approached. It has been observed in the various applications that bounded simulation based pattern matching generate more meaningful candidate matches than a structural based matching. It takes cubic time which is in $O(|V||E|+|EP||V|^2+|VP||V|)$ for computing bounded simulation relation which generates the exact matches in the data graph for a pattern graph. This is algorithmically better of graph simulation [4][8].

2) Incremental approach:In emerging applications, data graphs are large and dynamic in nature; data graph updations are frequent as operational data is frequently updated (insertions/deletions). Hence in pattern matching, it may result in recompilation of candidate matches from the scratch when data graph is updated. This limitation is overcome by incremental graph pattern matching approach, as this approach tries to find changes ΔM to the previously found matches M in response to changes in data graph ΔG. The computation of change value minimizes the match re-computation, as small change in data graph leads to smaller change in matches too. Hence, computing the degree of change is less than to recompute the entire matches for data graphs. In incremental approach, first step is to compute the matches on the entire graph via a batch matching algorithm and subsequently incrementally identifying new matches in response to change in data graph, ΔG [4].

Inputs for incremental matching technique are data graph, a pattern, the matches, and changes in data graph. The primary objective of the approach is to determine changes in matches (M) to the old candidate matches [6]. When a data graph is updated, technique evaluates new matches by using the previous computation of matches. Hence the costs of these approaches are defined as a function of the input size, similar to that of traditional complexity analysis. Instead, the algorithmic complexity must be analyzed in terms of |CHANGED|, which simply the degrees of change in the data graph.

*3) Strong Simulation approach:*Traditional graph pattern approaches, inflict topological constraint on data graphs to find relevant candidate matches for a pattern graph [9]. Technique in [4][19] faces trade-off between the low complexity and capability to retain the topology of data graph in the matches. As an outcome, multiple candidate matches of pattern P are generated having structure different from G. Strong simulation is capable to overcome this limitation by simply imposing two conditions in pattern matching. First, introducing the data duality to preserve the upwards edge-to edge mappings and second data locality to reduce the number of excessive candidate match set. The data duality and locality helps to capture the matched topology in data graph and thus enhances quality of candidate matches. Strong simulation extends the duality in the process to capture the graph topology and preserve the parent relationship similar to the bounded simulation [11[19]]. Strong simulation has cubic time complexity in capturing the candidate matches; it is equivalent to the bounded simulation. Dual simulation is a specific variant of strong simulation, which preserves both parent and child relationship. Both, strong and dual simulations are based on the concept of data locality in the data graph with respect to pattern query.

*4) View based approach :*Views are efficiently used for querying relational and semi-structured data. View based pattern matching is important and relevant notion for query processing in data integration systems, data mining and warehousing systems etc. When data graph is accessed via a set of views; pattern containment is derived for answering query pattern. It positions that a query mapped into graph pattern query can be answered by using a set of views, if pattern query is enclosed in a set of views [5][17]. When using views in graph pattern matching for a pattern P, a set of views V $=\{V_1,..V_n\}$ has to be defined on data graph G. Indeed, if materialized pattern views are used to compute answer to pattern query. Then there is only need to explore the views in the view set, instead accessing the original data graph and views can be easily merged to answer graph pattern query.

In the existing literature, there are two ways to pattern matching using views: Query Answering and Query rewriting [17]. Query answering simply compute query result by evaluating equivalent query which only refers to the views present in the view set. While, Query Rewriting reformulate the user query into another equivalent query which only refers to views present in the view set. In below table, we have consolidated some of the recent research effort in this direction and highlighted the various aspects of works, also listed in table 1.

TABLE I. TABLE 1: BRIEF SURVEY ON RECENT WORK OF GPM ON QUERY PROCESSING

Research Topic , Author(s)	Problem Discussed and Proposed Solution	Strengths	Limitation/Scope
Subgraph Isomorphism	**Problem-** Subgraph isomorphism **Solution-** Find the subgraphs isomorphic to pattern or returns set of subgraph isomorphic to the pattern with the best matched nodes.	• Finds exact matches. • Uses data locality. • Preserves adjacency and data graph topology.	• Unable to find a node pair using an arbitrary length path. • Search space increases exponentially with graph size. • Matching is neither bounded nor semi-bounded. • Intractable problem • Impose a strict graph topological constraint • Infeasible to find candidate matches in large and dynamic data graphs.
Graph Simulation	**Problem-** Constraint Matching **Solution** - Finds exact matches for the pattern graph	• Edge to edge mapping. • Bounded	• Impose a strict topological constraint • Does not pose data locality. • Does not captures graph topology of patterns in its matches
Bounded Simulation	**Problem-**This is simply based on connectivity of node pair connected via an arbitrary length of path. **Solution-** Each data graph node uses a pattern search conditions, search is based on contents or label of the node.	• Edge to path mapping • Bounded • Finds maximum bounded simulation relation rather than bijective functions.	• Outperforms for cyclic patterns.
Incremental Approach	**Problem**- Finding matching when data graphs are frequently updated with small changes **Solution-** Pattern search is based on the connectivity of node pair in predefined number of hopes in the dynamic data graphs with pattern graph.	• Semi bounded. • Reuse previous computations. • Complexity is given in terms of the area affected by updates.	• Outperform the counterparts in pattern matching on dynamic data graph. • Unbounded for matching based on subgraph isomorphism. • Cyclic patterns are still a problem
Strong Simulation	**Problem-** Capturing the topology of data graphs **Solution-** Preserves the topology of graphs matched such as parent, connectivity, cycle, by enforcing the graph duality and data locality on data graph for the candidate pattern matches.	• Uses data locality • Find bounded number of matches • Due to data locality, effective in pattern matching on distributed systems	• Unable to response in the large and dynamic data graph.
View based approach	**Problem-** Querying semi structured data and finding answer to queries using views without accessing the large social graphs **Solution-** The pattern containment is introduced for finding the candidate matches for a pattern query, according to the view sets. View sets are defined over frequent queries on the DB.	• Can be easily extended to edge-label graphs and queries • Combined with existing distributed, compression and incremental techniques for graphs	• To decide which view/ view set to cache for result generation. • Views initiation is problem. • View maintenance is a problem.

III. PERFORMANCE STATSTICS

In following some the comparative observation are discussed over the various GPM approaches,

- Subgraph isomorphism is often used in application domains in which identical candidate pattern matches are desired [7]. Though the complexity of subgraph isomorphism makes it reasonable to determine the candidate matches in “big” data graphs. Although this approach sometimes suffers to determine the ‘inexact’ matches. Moreover, variants of bounded simulation, such as strong or dual in which topology is preserved

[9], may induce results that "approximate" isomorphic subgraph.

- In disparity to the NP-hardness of subgraph isomorphism, bounded simulation based pattern matching is a cubic-time notion. Compared to graph simulation which takes quadratic time, bounded simulation is simpler as: Pattern graph P is classically much smaller than data graph G and $|E|$ is *in* $O(|V|^2)$ in the worst case.
- Graph pattern matching using incremental matching is semi-bounded and is constrained by the size of the changes in the input, output, and size of graph patterns and some auxiliary computations which computed from the previous runs. While graph pattern matching according using subgraph isomorphism is neither bounded nor semi-bounded.
- The incremental algorithms significantly do better than their batch counterparts in small changes in dynamic data graph. In modern day computing, dynamic and huge data graph is a reality, thus incremental pattern matching approach are promising as most of the real life graph are dynamic.
- The notion used for strong simulation can be adapted for the bounded simulation on regular candidate pattern graphs. The performance of strong simulation is cubic-time rather than NP-complete, due to the notion of data locality. Cardinality of candidate matches is linearly related with size of the data graph rather than exponential, where each candidate pattern match is constrained by the diameter of pattern graph.
- As opposed to graph simulation, strong simulation moreover captures the pattern's topology in candidate matches. The pattern's topology is represented by parents, connectivity and cycles, by enforcing the duality and locality on candidate matches, while keeping the complexity measure same as in the simulation. Unlike in the graph simulation, strong simulation's locality helps to improve the pattern matching on distributed graphs efficiently.
- In subgraph isomorphism [13] and strong simulation [9] graph pattern matching is according to the data locality in the data graph. Data locality makes distributed query evaluation easier since only a bounded or limited number of sites may have to be visited.
- View based GPM techniques can be voluntarily adapted for data graphs and pattern queries with edge labeled. Indeed, an edge-labeled data graph can be changed to a node labeled data graph: for each edge on data graph e, add a "dummy" node carrying the edge label of e, along with two unlabeled edges.

Algorithmic complexity of GPM, with the complexity of computing the cardinality of $M(P,G)$, for data graph $G = (V, E, f_A)$ and pattern graph $P = (V_P, E_P, f_v, f_e)$ of various approach, shown in table 2.

TABLE II. ALGORITHMIC COMPLEXITY OF GPM APPROACHES

Matching Approach	**Complexity**	**Cardinality**
Subgraph Isomorphism	$O((|V|^2+|V_P|)(|E|+|E_P|))$	$O(|V|^{|V_P|})$
Graph simulation	$O((|V|+|V_P|)(|E|+|E_P|))$	$O(|V||V_P|)$
Bounded simulation	$O(|V||E|+|E_P||V|^2+|V_P||V|)$	$O(|V||V_P|)$
Incremental approach	$O(|\Delta G|(|P||AFF|+|AFF|^2))$	$O(|V|)$
Strong simulation	$O(|V|(|V|+(|Vq|+|Eq|)(|V|+|E|)))$	$O(|V|)$
View based approach	$O(|P_S||\mathcal{V}(G)| + |\mathcal{V}(G)|^2)$	$O(|V|)$

IV. CHALLANGES

In recent years, lot of research contribution has been made by researchers from the various application domains on pattern matching. The classification of various pattern matching approaches is according to the semantics and structures. There are various inherent challenges to each of the category, such as indexing of candidate matches, candidate selection, and identifying matching. In below subsection, we are discussing some additional challenges for graph pattern matching:

- **Data Volume, Velocity, Variety and Veracity:** Evolution of various social medium such as Facebook, twitter etc., and the various data aspect emerges as challenging forces. Data is generated in tremendous rate by various computing devices in huge amount (in Zeta bytes) and the nature of data generated is complex (structured, semi-structured and unstructured)[15][16]. In real world there are various applications which are dealing with such big data for processing and analytics, however there are a few globally acceptable solutions are designed.
- **Query Access Pattern**: In most of the traditional application user query are static and simple over graph database. In modern data processing, user query are significantly dynamic and complex in nature as predicates in query are complex and relevant data sources are too many for a user query.
- **Index adjacency**: For implementation aspect, graph database is stored using index adjacency. In index adjacency, the relationships among data items are stored in index list while connected data graph nodes are physically stored in any other form of database (Relational/XML).
- **View Selection and Maintenance**: In view based pattern matching approaches, it is required to have the relevant and updated view for answering pattern queries. Selection of relevant view/views is a complex task and maintaining the data records in views (in materialized views) is also imposing a challenge [18].
- **Scale/Up gradation of existing graph:** In case of any expansion or extension, the entire data graph is required to be realigned and pattern query need to be tuned accordingly [18].

V. CONCLUSION AND FUTURE SCOPE

Finding a match to a user query in data graph is a prominent problem in data mining and evolution of social media analytics and mining is one of the prime areas of application. The dilemma of pattern searching in data graph or graph-structured data has many applications areas. Hence, several efficient graph pattern matching techniques have been developed in past few years. Mutually, these matching techniques signify decades of work by researchers from a diversity of research communities. Regardless of variations in inherent properties of graphs, data sets, and algorithms, common premise have emerged. This paper briefly summarized the various approach of graph pattern matching irrespective of the application areas and highlighted the possible challenges in modern day computing. Some of the prime challenging forces in modern day computing, like increasing data volume and heterogeneity in nature of data, extensibility of computing network etc. further we discussed the possible future direction in the existing approaches by keeping the work done in the area of data query processing and suggested some of the areas of future directions.

Based on literature review, we have identified some of the observations as future directions in the graph pattern matching,

- By considering the more efficient batch and incremental matching along with advanced features, such as graph indexing, summarization and data compression techniques, lot of improvement on the lower bounds of graph pattern matching can be achieved.
- In the existing research it is observed that, multiple researchers used an incremental approach based graph pattern matching approach for patterns of the form DAGs, but same is not yet applied for cyclic pattern with assured good performance.
- Advanced ranking mechanism can be adapted for ranking candidate matches in strong simulation. The ranked top-k matches are retrieved and further the same can be adapted for regular expressions on edge types.
- In view based pattern matching approaches, deciding on which view or views are relevant and to be cached for answering frequent used pattern queries is complex problem. Thus an efficient mechanism can be designed based on the subgraph isomorphism.
- All the existing views based graph pattern matching techniques are purely according to the containments of pattern in data graph. Thus an efficient algorithm for computing the maximal containment is required for the scenario, when containment is not present in available view set.

REFERENCES

[1]. Angles, R., Gutierrez, C.: Survey of Graph Database Models. ACM Computing Surveys, Vol. 40, No. 1, Article 1 (2008)

[2]. Bhargavi, B., Supreethi, K.P.: Graph Pattern Mining: A Survey of Issues And Approaches. International Journal of Information Technology and Knowledge Management, Vol. 5, No. 2, pp. 401-407 (2012)

[3]. Fan, W.: Graph Pattern Matching Revised for Social Network Analysis. ICDT 2012. Copyright 2012 ACM 9781450307918/ 12/03(2012)

[4]. Fan, W., Li, J., Ma, S., Tang, N., Wu, Y., Wu, Y.: Graph Pattern Matching: From Intractability to Polynomial Time. Proceedings of the VLDB Endowment, Vol. 3, No. 1 (2010)

[5]. Fan, W., Wang, X., Wu, Y.: Answering Graph Pattern Queries Using Views. IEEE Transactions on Knowledge and Data Engineering, ICDE, pp 184-195(2014)

[6]. Fan, W., Wang, X., Wu, Y.: Incremental Graph Pattern Matching. ACM Trans. Datab. Syst. 38, 3, Article 18 ,47 pages (2013)

[7]. Gallagher, B.: Matching Structure and Semantics: A Survey On Graph – Based Pattern Matching. American Association for Artificial Intelligence (2006)

[8]. Henzinger, M. R., Henzinger, T., Kopke, P.: Computing Simulations on Finite and Infinite Graphs. Foundations of Computer Science, pp 453-462 (1995)

[9]. Ma, S., Cao, Y., Fan, W., Huai, J., Wo, T.: Capturing Topology in Graph Pattern Matching. Proceedings of the VLDB Endowment, Vol. 5, No. 4 (2012)

[10]. Natarajan, M.: Understanding The Structure Of A Drug Trafficking Organization: A Conversational Analysis. Crime Prevention Studies, Vol. 11, pp 273–298 (2000)

[11]. R. Milner. Communication and Concurrency. Prentice Hall, 1989

[12]. Robinson, I., Webber, J., Eifrem, E.: Graph Databases. O'Reilly (2013)

[13]. Ullmann, J. R.: An Algorithm for Subgraph Isomorphism. J. ACM, Vol. 23, No. 1,pp 31-42 (1976)

[14]. Zou, L., Chen, L., Ozsu, M.T.: Distance-Join: Pattern Match Query In a Large Graph Database. VLDB Endowment, ACM 978-1-60558-948-0/09/08 (2009)

[15]. Nisar, M. Usman, Arash Fard, and John A. Miller. "Techniques for Graph Analytics on Big Data", 2013 IEEE International Congress on Big Data, 2013.

[16]. Miller, John A., Lakshmish Ramaswamy, Krys J. Kochut, and Arash Fard. "Research Directions for Big Data Graph Analytics", 2015 IEEE International Congress on Big Data, 2015.

[17]. Fan, Wenfei, Xin Wang, and Yinghui Wu. "Answering Pattern Queries Using Views", IEEE Transactions on Knowledge and Data Engineering, 2015.

[18]. Ma, Shuai, Jia Li, Chunming Hu, Xuelian Lin, and Jinpeng Huai. "Big graph search: challenges and techniques", Frontiers of Computer Science, 2015.

[19]. Ma, Shuai, Yang Cao, Wenfei Fan, Jinpeng Huai, and Tianyu Wo. "Strong simulation: Capturing topology in graph pattern matching", ACM Transactions on Database Systems, 2014.