Chapter 6
A Survey of Relational Approaches for Graph Pattern Matching over Large Graphs

Jiefeng Cheng
The University of Hong Kong, China

Jeffrey Xu Yu
The Chinese University of Hong Kong, China

ABSTRACT
Due to rapid growth of the Internet and new scientific/technological advances, there exist many new applications that model data as graphs, because graphs have sufficient expressiveness to model complicated structures. The dominance of graphs in real-world applications demands new graph processing techniques to access and analyze large graph datasets effectively and efficiently. Among those techniques, a graph pattern matching problem receives increasing attention, which is to find all patterns in a large data graph that match a user-given graph pattern. In this survey, we review approaches to process such graph pattern queries with a framework of multi joins, which can be easily implemented in relational databases and requires no specialized native storage for graphs. We also discuss the top-k graph pattern matching problem.

INTRODUCTION
Graph structured data is enjoying an increasing popularity as Web technology and new data management and archiving techniques advance. Due to the sufficient expressive power of graphs to model complex relationships among objects, numerous emerging applications need to work with graph data. Instances include navigation behavior analysis for Web usage mining (Berendt & Spiliopoulou, 2000), web site analysis (Fernandez, Florescu, Levy & Suciu, 1997), biological network analysis for life science (Helden et al., 2000) and so on. The dominance of graphs in real-
world applications demands new graph processing techniques to access large data graphs effectively and efficiently. In 2002, Shasha, Wang, and Giugno highlighted algorithms and applications for searching trees and graphs.

Among those techniques, a graph pattern matching problem receives increasing attention. Traditionally, graph pattern matching usually stands for problems like the subgraph isomorphism (Ullmann, 1976), which is long investigated. This problem is widely used recently for subgraph containment search (Yan, Yu, & Han, 2004; He & Singh, 2006; Williams, Huan, & Wang, 2007; Cheng, Ke, Ng, & Lu, 2007; Zhao, Yu, & Yu, 2007; Shang, Zhang, Lin, & Yu, 2008). They often work with large number of small candidate graphs with hundreds or thousands of nodes. However, in many cases, the database consists of only one or several very large graphs, such as protein-protein interaction networks, Web information, social networks, etc. Urged by the need to efficiently manage and analyze very large graphs, new query semantics for graph pattern matching over very large graphs are developed beyond the subgraph isomorphism semantics: a small graph pattern \( Q \) is used to represent structural requirements, in which the connectedness between matched nodes are emphasized. To be more specific, \( Q \) implies two kinds of conditions that the matched nodes in \( G_D \) should satisfy: (1) The labels conditions specified by \( Q \) request that all matched nodes in \( G_D \) are labeled identically with \( Q \); (2) the reachability conditions of \( Q \) suggest all matched nodes in \( G_D \) should have connected paths in \( G_D \) which are required to match all edges in \( Q \).

There are many applications for such graph pattern matching. Taken DBLP as an example (Figure 1(a)). The XML document can be considered as a large graph by categorizing papers into different conferences/journals followed by years. Different papers may link to the same author, which exhibits a graph structure. A simple graph query (Figure 1(b)) finds the authors who have papers published at the major database conferences, namely, EDBT, ICDE, SIGMOD, and VLDB, in the same year. The answer set includes Christos Faloutsos who did it in 1994, and Hector Garcia-Molina and Surajit Chaudhuri who did it in 1996. In this example, a graph pattern queries the graph-structured XML document, by extending twig queries over the tree-structured XML document (Bruno, Koudas, & Srivastava, 2002). In biological networks such as protein-protein interaction networks (PPI) and metabolic networks, a small graph pattern \( Q \) can represent a pattern of some kind of interactions or pathways which is interested by scientists, and can be used to further search similar patterns in exploring different data sets. In software systems, a small graph pattern \( Q \) can be used to find interesting dependency patterns in dependency graphs.

One question which may be asked by keen readers is that in connected undirected graphs or strongly connected directed graphs, any node is reachable from (or reaches to) any other node which automatically satisfies the reachability conditions of a given graph pattern query. To explain that our graph pattern matching problem is still useful, we can see that in real applications, many XML graphs and biological networks are still very sparse with the edge/vertex ratio slightly larger than 1 (Wang, He, Yang, Yu, & Yu, 2006); additionally, to handle highly cyclic graphs or even undirected graphs, selection constraints can be deliberately added to prune those reachable nodes which are not needed. For example, in analyzing online social networking systems, such as Facebook, a large graph can be obtained where the “job-title” attribute on each node is regarded as the node label. A small graph pattern \( Q \) can be used to discover connections between several people with specified jobs. However, in general, it can return many nodes with obscure connections among them. Therefore, in 2009, Zou, Chen, and Özsù employ a distance threshold to prune all connected nodes whose distances are larger than the threshold. For example, In Figure 2(a), the node-tuple \( \langle 3, 5, 6, 8 \rangle \) matches \( Q \) in Figure 2(b)