International Journal of Semantic Computing
Vol. 12, No. 4 (2018) 595–608
©World Scientific Publishing Company
DOI: 10.1142/S1793351X18500071

Technology Outlook

GOLAP: Graph-Based Online Analytical Processing

Chung-Hsien Chou[*,‡], Masahiro Hayakawa[†], Atsushi Kitazawa[†]
and Phillip Sheu[*]

[*]*Department of Electrical Engineering and Computer Science*
University of California, Irvine, CA 92697, USA

[†]*NEC Solution Innovators, Ltd.*
1-18-7 Shinkiba, Koto-ku, Tokyo 136-8627, Japan
[‡]*chunghc3@uci.edu*

Graph-based Online Analytical Processing (GOLAP) extends Online Analytical Processing (OLAP) to address graph-based problems that involve object attributes. Based on graph data, GOLAP can answer user queries related to combinatorial optimization, structural analytics, and influence analytics. Besides, since a GOLAP system is an online interactive system that requires fast response time, the execution time for graph-problem queries is essentially critical. Thus, how to speed up the execution time of specific graph problems becomes a challenge in GOLAP. In this paper, we show several methods to speed up the running time, including graph data reduction and approximation. In this paper, we survey classes of graph-based queries, challenges for GOLAP, and solutions that GOLAP provides.

Keywords: GOLAP; graph; approximation; graph data reduction.

1. Introduction

Graph theory dates back to 1735, when Leonard Euler solved the Seven Bridges of Königsberg problem: “Is it possible to design a walking tour of the city in which you cross all of the seven bridges exactly once?” Euler designed a mathematic model consisting of vertices and edges to solve this problem, laying the foundation of graph theory. Since then, graph theory has found many use cases. One of the famous questions is the “Four-Color Problem” posed by Francis Guthrie in 1852, but the problem was not solved until 1969. Heinrich Heesch solved the problem with the assistance of a computer. Although graph theory has been researched for a long time, only recently has it been applied to storing and managing data [9].

One major application of relational data is called Online Transactional Processing (OLTP), which is efficient in running a large number of simple transactions. In addition to OLTP, Online Analytical Processing (OLAP) has been popular for fast

[‡] Corresponding author.

and multi-dimensional data analytics. On the other hand, it is widely recognized that many complex data can be stored as graphs, such as gene regulatory networks, social networks, and road map networks. However, there is a lack of comprehensive online Graph-based OLAP (GOLAP) systems that can answer user queries regarding graph problems. Because it often requires fast response to online user queries, how to speed up execution time for graph problems becomes a challenge in GOLAP. Another challenge is memory space, especially when dealing with large graphs. In this paper, we survey classes of graph-based query problems, challenges for GOLAP, and solutions that GOLAP provides. The classes of graph-based query problems in GOLAP include combinatorial optimization problems, structural analytics, and influence analytics.

In Sec. 2, we define the three dimensions in GOLAP: classes of queries, challenges, and solutions. Since a GOLAP system is interactive that requires fast response time, the execution time for any class of graph-problem query is essentially critical. Therefore, how to speed up the running time of specific graph problems is a challenge in GOLAP.

In Sec. 3, we survey the existing researches related to GOLAP including graph database, structural analytics, graph data reduction, and approximation.

2. GOLAP

Chen *et al.* [12] proposed the basic definitions of operations in GOLAP, using a simple example related to authors and conferences. Figure 1 shows the graph-based scenario for a conference example. Given a set of authors working in a field, if two authors coauthor one or more papers in a conference, they will be linked together with a weight w which is the number of papers coauthored. For every conference in every year, they may have a corresponding coauthor network in which they show the relationship between two authors. Each of the networks is a subgraph of the overall (aggregated) coauthor network. For example, one may want to check the collaboration patterns of each conference (including SIGMOD, VLDB, ICDE, etc.) in each year. In the language of data cube in graph-based OLAP, there are two dimensions in this example, venue and time. One may be interested in the collaboration for the venue SIGMOD in 2004, therefore, one can apply drill-down process to obtain all the subgraphs for each venue in each year from the aggregated graph. Similarly, one can choose the subgraph for SIGMOD 2004. On the other hand, one may need to know the overall collaboration for all the authors in all venues in all years, so he can apply a roll-up procedure to obtain the aggregated graph. In the roll-up procedure, we may need to modify the weights on the links by summing up all the respective edge weights as shown in Fig. 1.

We may formalize GOLAP by assuming that each node in a graph is associated with one or more categories. In our definition, categories in GOLAP are like multiple dimensions in traditional OLAP. When a roll-up is executed, we can view/query the graph at the class level. As classes may be hierarchically structured, another roll-up will allow the user to view/query the graph at the superclass level. This process can

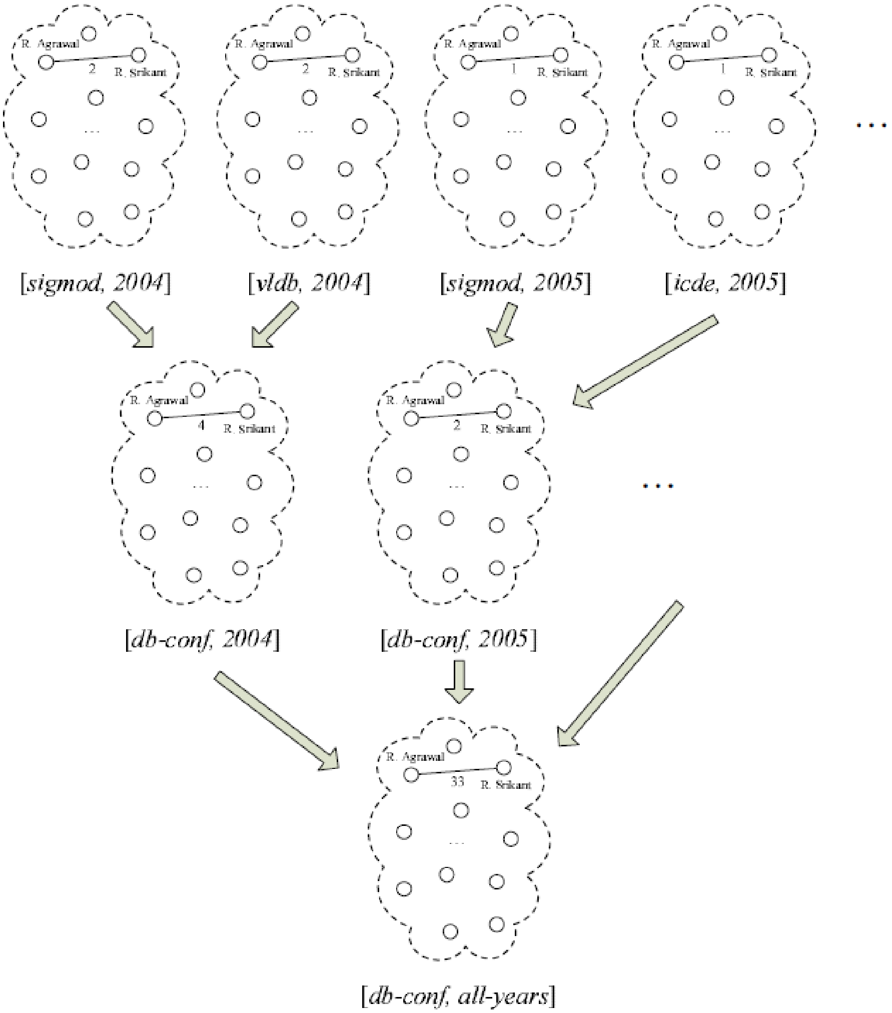


Fig. 1. Two process cubes illustrating the splitting (drill-down) and merging (roll-up) of process cells using the case type and event class dimensions [12].

be repeated until the highest level of abstraction is reached. Drill-down operations will reverse the above that allows the user to view/query the graph with more details. Not only the nodes of a graph may be classified, the links of a graph may be classified as well. Therefore the user may choose to roll up along the node hierarchy or the link hierarchy. In a roll-up process, all the nodes that belong to the same class may be combined into one node. After a drill-down, the details about all the nodes inside each node are revealed. Consider the example shown in Fig. 2, after a roll-up, all the genes related to breast cancer (share the same color green) are combined into a node called "Breast Cancer". A drill-down process breaks the "Breast Cancer" node into details which are four genes "Gb1," "Gb2," "Gb3," and "Gb4." Thus, for this version

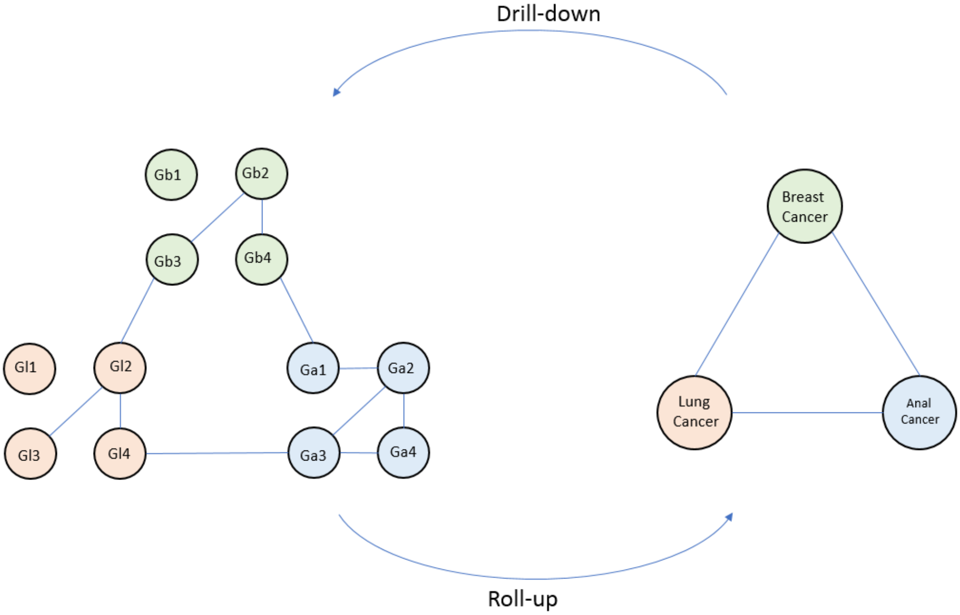


Fig. 2. A genetic example for the proposed roll-up and drill-down operations.

of roll-up and drill-down, GOLAP can answer user queries based on categorized graph data at summary level. Figure 3 shows another example for drill-down and roll-up in a graph containing information among IT companies in different states. After a roll-up process, the edge weight between the companies in CA and the companies in NY can be calculated as the sum of the weights on the edges or paths between all pairs of companies from the two states. For example, the weight on the edge from node "IT in CA" to node "IT in NY" is the total sum of weights on the paths from node "A Co." to node "G Co.", from node "A Co." to node "H Co.", from node "B Co." to node "G Co.", and from node "B Co." to node "H Co.". Another class-oriented query problem that can be answered in our GOLAP is "find the company in California who has the strongest influence on all the other companies in New York state." After the drill-down process, in the detailed graph, we can calculate the influence factor of all the pairs from each company in California to each company in New York. And once the company with strongest influence to all companies in New York is found, we can take the result as a summary of the strongest influence from companies in California to those in New York for the class-oriented graph (after roll-up).

On the other hand, for the link hierarchy, every link between nodes can also be classified and has a label or color that represents the same category. Therefore, with the two sets of dimensions which are dimensions for nodes and dimensions for links, respectively, we can answer various combinatorial queries, with respect to the concepts of colored/labeled graph problems. For example, in Fig. 4, there are nodes representing customers and links representing common interests between customers,

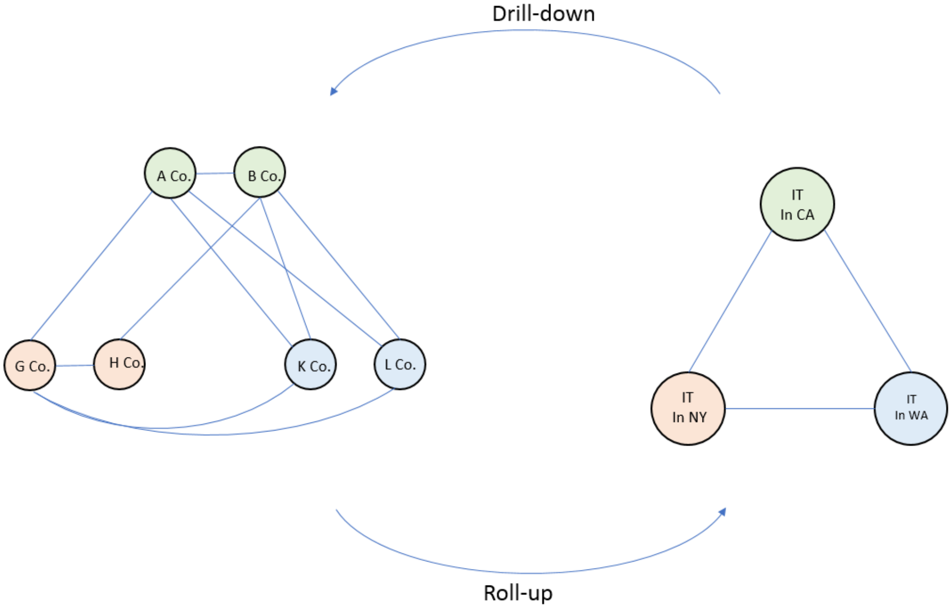


Fig. 3. An example for the proposed roll-up and drill-down operations.

and there is one set of dimensions for nodes and another set of dimensions for links. The customers are classified using colors on the nodes corresponding to the states where they reside, while the colors on the links mean the common type of products that the two customers share interest on. Thus, with the combination of two sets of dimensions, we can post queries such as "Find the customer in California that has the highest influence on all the other customers who share the same interest with him/her."

In the research area related to GOLAP, most researches have different definitions for the OLAP operations: consolidation (roll-up), drill-down, and slicing and dicing. In addition, they develop cross-disciplinary applications focusing on how to find out

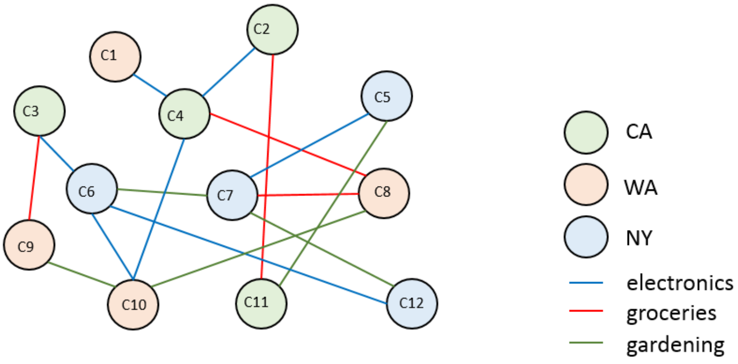


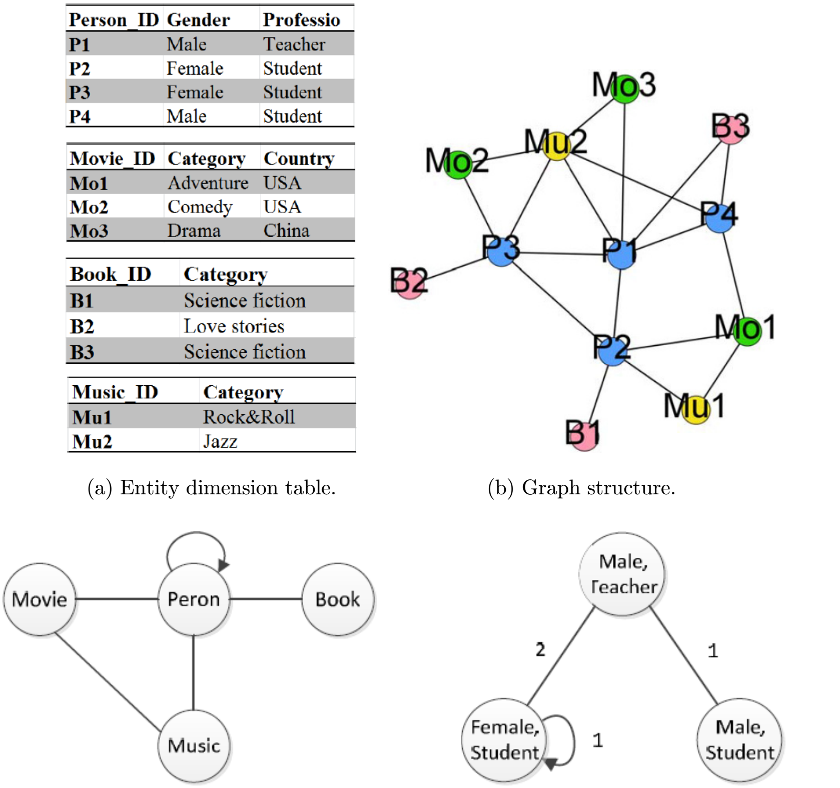
Fig. 4. An example for node hierarchy and link hierarchy.

consolidated summary of the information retrieved from the graph data. In the following paragraphs, we list some of these GOLAP researches.

Beheshti *et al.* define a semantic framework and a language to support graph-based OLAP. Although some research has already defined graph-based multi-level and multi-dimensional graph-based queries, there is not much work devoted to semantic-driven query language like those in traditional OLAP. They extend SPARQL to support graph-based queries, but they only focused on solving partitioning problem. For example, for Fig. 1, their query language can support the following query: "partition the original graph into a set of subgraphs corresponding to diseases." Our extended definition of GOLAP considers the two dimensions of nodes and edges, and therefore we consider new queries that are enabled by these dimensions. Taking an example in Fig. 4, with our GOLAP, we can answer a query: "For all the customers who have common interest in electronic products, find the customer in California State that has the highest influence on all the customers in New York State."

Wang *et al.* [11] proposed Pagrol over attributed graphs, such as the Web and various social networks (e.g. Facebook, LinkedIn, Twitter), which is a parallel graph OLAP system. Moreover, in order to aggregate attributed graphs at different granularities and levels, they introduced a conceptual Hyper Graph Cube model, which is an attributed-graph analog of the data cube model for relational database management systems. Their model also contains two sets of dimensions for nodes and edges, and Pagrol provides functionalities including posting queries of graph-version roll-up and drill-down operations. However, they did not focus much on solving queries for combinatorial problems using two-dimensional GOLAP. For example, they did not address the queries like: "Find the top five customers in all states that have the strongest influence on the customers who share the same interests in gardening."

Another graph cube framework which is called Two-Step Multi-dimensional Heterogeneous (TSMH) was proposed by Wang *et al.* [13]. In order to guide the aggregation of the network and build the Entity Hyper Cube, the meta path was used in heterogeneous network. They also designed meta path aggregation algorithms and proposed the materialization strategy. In dimension cube, they significantly improved the efficiency of dimension operations using hierarchical coding for entities and dimensions which saves the overhead of join operations of entities and dimensions. For the example in Fig. 5 [13], there are four types of entities in Fig. 5(a), and each entity has different number of attributes on node-dimension, so that the corresponding network in Fig. 5(b) is called a heterogeneous network. They proposed roll-up/drill-down operations for heterogeneous networks, in order to obtain a summarized graph as shown in Fig. 5(c) or an aggregated graph with respect to specific attributes as shown in Fig. 5(d). However, they focused more on aggregating different attributes, but they did not extend the work for more complicated query problems, such as: "Find the student that likes adventure movies and has the strongest influence on all the other persons".



Person_ID	Gender	Professio
P1	Male	Teacher
P2	Female	Student
P3	Female	Student
P4	Male	Student

Movie_ID	Category	Country
Mo1	Adventure	USA
Mo2	Comedy	USA
Mo3	Drama	China

Book_ID	Category
B1	Science fiction
B2	Love stories
B3	Science fiction

Music_ID	Category
Mu1	Rock&Roll
Mu2	Jazz

(a) Entity dimension table.

(b) Graph structure.

(c) Schema of the network.

(d) Aggregate graph on gender and profession.

Fig. 5. A sample multi-dimensional heterogeneous network [13].

Another cross-disciplinary research plays an important role between data science and marine studies. Du and Shao [10] proposed an evaluation method for seawater quality. It applies GOLAP composed of seawater quality data to develop a multi-dimensional analytics approach. And it calculates the differences of water quality between different areas and constructs monitored areas network. Nowadays, studies on seawater quality are essential when solving the global seawater issues, such as marine eutrophication and prediction of marine algal blooms. Apart from conventional studies which usually take statistics data and apply machine learning methods, the authors show that GOLAP is an efficient way for marine data. Besides, the water quality distance threshold they proposed is useful when evaluating marine eutrophication and predicting marine algal blooms.

GOLAP also contains various classes of queries, including combinatorial algorithms, structural analytics, and influence analytics. Users can post diverse types of query problems against graph databases, such as shortest paths, minimum spanning tree, reachability, etc.

3. GOLAP Queries

In this section we summarize several classes of queries that may be posed in our GOLAP system.

3.1. *Combinatorial optimization*

One class of GOLAP queries is combinatorial problems. Graphs are used to represent networks of communication, data organization, computational devices, the flow of computation, etc. For example, the link structure of a website can be represented by a directed graph, where the nodes represent web pages and directed edges represent links from one page to another. A similar approach can be taken to problems in travel, biology, computer chip design, and mapping the progression of neurodegenerative diseases [42]. There are numerous graph combinatorial problems, for example, four-color theorem [43], Hamiltonian path problem [44], minimum spanning tree [45], Chinese postman problem [46], Seven Bridges of Königsberg [47, 48], three-cottage problem [49], traveling salesman problem [50], and max-flow min-cut theorem [51]. Taking an example in Fig. 4, one possible query for the combinatorial optimization problems in terms of shortest path problem in GOLAP is: "Find a customer in New York State that has the most similar preference on electronic products with customer C5."

3.2. *Structural analytics*

Another class of queries in graph databases is the structural analytics that has been well researched. Structural analytics in GOLAP focuses on the aggregation of data which is stored in the vertices or edges in the graph databases. Also, structural analytics in GOLAP deals with the graph structure itself, including finding the quantitative and qualitative centers in the graph, such as betweenness centrality and closeness centrality. Betweenness centrality finds the node that could represent the graph network or be the backbone of the graph network. The backbone has the most numbers of shortest paths passing through among the whole network. Closeness centrality finds the node that has the most central position which is closest to all the other nodes in the network. Another important dimension of semantic queries in GOLAP which is structural analytics on graphs has been well studied, especially in the social networks domain. Many of the graph structural measurements can be applied to graph databases. We briefly describe some of the measurements here.

One of the popular topics is centrality, which means identifying the most important nodes (vertices) within a graph. Freeman [21] presented an important conceptual review of centrality measures for dichotomous network data; "degree-based" measures focus on the degree of communication activity; "betweenness" measures the frequency of a node being between communications of others; and "closeness" measures reflect freedom from the control of others. Gould [22] extended Freeman's betweenness measures to nonsymmetrical data. Stephenson and Zelen [23] gave a related centrality measure based on information. Everett and Borgatti [24] experimented with the relationship between ego network betweenness (i.e. a local property) and betweenness of the node in the entire network. They found that, while there is no formal connection, there is a high correlation in many real-life and randomly generated graphs. Rattigan *et al.* [25] used structure indices of the network to efficiently approximate betweenness and closeness centralities. Another interesting measurement is tie strength, which means to identify the strength of a relation (edge). Granovetter [26] introduced the concept of tie strength and proposed four tie strength dimensions: amount of time, intimacy, intensity, and reciprocal services. Burt [27] proposed that structural factors like network topology and informal social circles shape tie strength. Lin *et al.* [28] showed that social distance, embodied by factors such as socioeconomic status, education level, political affiliation, race, and gender, influences tie strength. Marsden and Campbell [29] used survey data from three metropolitan areas to precisely unpack the predictors of tie strength. Gilbert and Karahalios [30] presented a predictive model that maps social media data to tie strength. For the example shown in Fig. 2, with the two dimensions in GOLAP, we can post queries using betweenness centrality like: "Find the backbone (most important) gene in the network among all the breast cancer genes, lung cancer genes, and anal cancer genes."

3.3. *Influence analytics*

The other class of queries in GOLAP is influences analytics. In recent years, influence research in a social network has been a hot topic. In the US election in a social network, a democrat or a republican can influence another democrat or a republican on Twitter, or Facebook. In the area of science, an author/scientist may influence another author or a scientist. In influence graph model, edge represents a relationship between nodes, and weight of edge represents the strength of what the edge represents [51]. For influence analytics, with the merit of two dimensions in GOLAP, one possible query that GOLAP can answer in Fig. 4 is: "Find the customer in California State who has the strongest influence on all the other customers who share the same preference in buying groceries and live in California State."

4. Challenges in GOLAP

The challenges in GOLAP are speed and storage. When the graph data size becomes really large, such as a graph containing millions of nodes and millions of edges, the execution time $O(n^2)$ of an algorithm could be very long. Since GOLAP must answer user queries online, it is not tolerable for users to wait for results for a long time. Furthermore, when dealing with large-size data, it usually requires considerable memory space. However, requiring hundreds of GBs of memory to run queries is a waste of memory space, and it is not practical to prepare such size of memory.

Possible solutions for the challenges include the following:

4.1. *Speedup*

One solution for the challenges is speedup. Sometimes, query problems require more execution time when dealing with large size of graphs (in terms of number of edges and number of nodes). Since GOLAP must answer user queries online, it is not tolerable for users to wait for results for a long time, such as hours or even days. Therefore, speedup is often essential when dealing with large graph networks or when the query problem does not run in linear time or even takes exponential time to execute. Thus, some approaches apply graph data reduction to speed up for these kinds of query problems.

4.2. *Graph data reduction*

When the GOLAP is dealing with large graphs, it requires a considerable memory space to load the graph data. If the graph is too large for the GOLAP to load it and execute semantic queries on it, one possible solution is applying graph data reduction. Since the graph data reduction algorithms reduce the data size by removing irrelevant edges or nodes preserving the query result, there is a possibility that GOLAP can load the graph and answer the user queries. Also, the data reduction algorithms can improve the performance since the graph size is reduced.

On top of syntactical reduction methods that may be applied to all applications (e.g. social networks, map applications, real-time recommendation engines), we can consider the specific problem(s) addressed by an application to reduce a graph further. Indeed, for many applications there is a lot of useless information in graphs, especially in social networks [4–6]. For example, one may only be interested in the relationships in a social network, but not concerned about what a person says. Therefore, by removing information irrelevant to an application we can improve the reduction ratio.

Such methods may be considered as semantic, or query-based, graph reduction methods (see e.g. [7, 8]). Many of the graph reduction methods in this category can be considered as "lossless" graph reduction [1] (e.g. online reduction [2] and dynamic online network reduction [3]) in the sense that after reduction we can obtain the same query results based on the reduced graph and the original graph.

Data reduction [31] is useful for many applications which involve big data. Existing approaches to graph data reduction may be syntactic or semantic, which may be lossless [32–34] or lossy [38, 39]. Lossy data reduction methods, also called lossy compression, for continuous sources have been well studied [41]. They are widely used for images, videos, and audios whose applications tolerate errors as human beings can still accept. Some of the approaches [35–37] use graphs in their algorithms for encoding and decoding the data. However, using the similar idea to reduce graph data in general is a new research direction, where only few publications can be found. Some works related to this direction are summarized in the following.

The reachability problem is a very important problem in graph computing. A lot of applications need to compute the results based on k-reachability. Reachability reduction methods [38–40] have been reported in some papers. While most researches on reachability reduction are lossless Zhou *et al.* [15] described a lossy network simplification method that reduces a graph with the minimum impact on graph connectivity. By maintaining the connectivity, most reachability and path queries can still be answered.

4.3. *Approximation*

We define another solution in GOLAP, which is approximation. When posting queries to the graph database, if the execution time is too long due to the enormous size of the graph, approximation algorithms in GOLAP can provide an approximated query result which requires less execution time with a tolerable error rate. Furthermore, approximation algorithms can deal with heuristics to conquer the original NP-hard query problems that require exponential time.

In order to achieve approximation where an error rate is controlled by predefined threshold, some researchers propose lossy reduction algorithms which can reduce the size of the graph and preserve the query result of the same semantic queries, while sacrificing the accuracy with a tolerable error rate.

While syntactically all graph reduction methods are lossy because the original graph is reduced to a smaller graph (therefore some data is lost), in this paper we interpret the term “lossy” more strictly, and semantically, referring to reduction methods that either lose the ability to answer some queries that may be answered from the original graph correctly or lose the correctness of some results.

Our hypothesis is that further reduction may be possible even on top of query-based reductions. Sometimes the accuracy of query results may not be strictly required. Specifically, if we allow approximate results, where the difference between the approximate results and the precise results is bounded, a graph may be further reduced.

The concept of lossy reduction in the following papers is based on graph structures, focusing on errors between the original graph and the reduced graph.

Navlakha *et al.* [19] described a graph summarization method with bounded errors. A graph can be reduced by a two-part representation: summary and corrections. The summary is an aggregated graph that groups the nodes into sets with edges representing relations between sets. The corrections part specifies a list of edge-corrections that should be applied to the summary to recreate the graph. The summarization can be lossless or lossy — here the term “lossy” means there are errors in finding the neighbors of each node. Toivonen *et al.* [20] compressed a weighted graph by merging nodes and edges to achieve the target reduction rate while minimizing the difference on the edge weights between the original graph and the compressed graph.

5. Conclusions

In this paper, we define the concept of GOLAP. We have shown the benefits using GOLAP in different disciplines, such as social network, biology, data science, and global pollutions. Besides, we have listed the common features in GOLAP, including semantic queries, structural analytics, speedup, graph data reduction, and approximation. These dimensions are considered blooming research topics, and they are continuously being studied by researchers.

We have surveyed the existing works on GOLAP which are focusing on definitions of graph-version OLAP operations, such as consolidation (roll-up), drill-down, and slicing and dicing. Some researchers also define methods used for query processing and optimization based on different graph database structures. In order to achieve a more efficient way compared to the other conventional methods, for example, using statistical programs or machine learning algorithms, some applications have been proposed by the researches using GOLAP.

One possible future direction is to develop more algorithms related to graph data reduction. Since now the proposed graph reduction algorithms are restricted to graph types in specific field, such as social science, we believe that more general algorithms could be developed for multiple fields. Another direction is to contribute to approximation dimension in GOLAP by developing more lossy data reduction algorithms and other heuristics, which can improve the required execution time when user accesses user-interactive queries on GOLAP.

Acknowledgments

The research is supported in part by NEC Solution Innovators, Ltd., Japan.

References

[1] P. Boldi and S. Vigna, The WebGraph framework I: Compression techniques, in *Proc. 13th Int. Conf. World Wide Web*, 2004, pp. 595–602.

[2] J. M. Hellerstein, P. J. Haas and H. Wang, Online aggregation, in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 1997, pp. 171–182.

[3] C. H. You, L. Holder and D. Cook, Graph-based data mining in dynamic networks: Empirical comparison of compression based and frequency-based subgraph mining, in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2008, pp. 929–938.

[4] F. Chierichetti, R. Kumar, S. Lattanzi, M. Mitzenmacher, A. Panconesi and P. Raghavan, On compressing social networks, in *Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2009, pp. 219–228.

[5] A. Clauset, M. E. J. Newman and C. Moore, Finding community structure in very large networks, *Phys. Rev. E* **70** (2004) 066111.

[6] P. Boldi, M. Rosa, M. Santini and S. Vigna, Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks, in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 587–596.

[7] W. Fan, J. Li, X. Wang and Y. Wu, Query preserving graph compression, in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 2012, pp. 157–168.

[8] H. Maserrat and J. Pei, Neighbor query friendly compression of social networks, in *Proc. 16th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2010, pp. 533–542.

[9] S.-T. Wang, J. Jin, P. Rivett and A. Kitazawa, Technical survey graph databases and applications, *Int. J. Semant. Comput.* **9**(4) (2015) 523–545.

[10] X. Du and F. Shao, Multidimensional analysis of seawater quality data based on graph OLAP, in *Proc. 8th Int. Symp. Computational Intelligence and Design*, Vol. 2, 2015.

[11] Z. Wang *et al.*, Pagrol: Parallel graph OLAP over large-scale attributed graphs, in *Proc. IEEE 30th Int. Conf. Data Engineering*, 2014.

[12] C. Chen *et al.*, Graph OLAP: Towards online analytical processing on graphs, in *Proc. Eighth IEEE Int. Conf. Data Mining*, 2008.

[13] P. Wang, B. Wu and B. Wang, TSMH Graph Cube: A novel framework for large scale multi-dimensional network analysis, in *Proc. IEEE Int. Conf. Data Science and Advanced Analytics*, 2015.

[14] X. Wu, B. Wu and B. Wang, P&D Graph Cube: Model and parallel materialization for multidimensional heterogeneous network, in *Proc. Int. Conf. Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2017.

[15] F. Zhou, S. Malher and H. Toivonen, Network simplification with minimal loss of connectivity, in *Proc. IEEE Int. Conf. Data Mining*, 2010.

[16] J. X. Yu and J. Cheng, Graph reachability queries: A survey, in *Managing and Mining Graph Data*, Advances in Database Systems, Vol. 40 (Springer, Boston, 2010), pp. 181–215.

[17] H. Yildirim, V. Chaoji and M. J. Zaki, GRAIL: A scalable index for reachability queries in very large graphs, *VLDB J.* **21**(4) (2012) 509–534.

[18] H. Wei, J. X. Yu, C. Lu and R. Jin, Reachability querying: An independent permutation labeling approach, *VLDB J.* **27**(1) (2014) 1191–1202.

[19] S. Navlakha, R. Rastogi and N. Shrivastava, Graph summarization with bounded error, in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 2008, pp. 419–432.

[20] H. Toivonen, F. Zhou, A. Hartikainen and A. Hinkka, Compression of weighted graphs, in *Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2011, pp. 965–973.

[21] L. C. Freeman, Centrality in social networks conceptual clarification, *Soc. Netw.* **1**(3) (1978) 215–239.

[22] R. Gould, Measures of betweenness in non-symmetric networks, *Soc. Netw.* **9** (1987) 277–282.

[23] K. Stephenson and M. Zelen, Rethinking centrality: methods and examples, *Soc. Netw.* **11** (1989) 1–37.

[24] M. Everett and S. P. Borgatti, Ego network betweenness, *Soc. Netw.* **27**(1) (2005) 31–38.

[25] M. J. Rattigan, M. Maier and D. Jensen, Using structure indices for efficient approximation of network properties, in *Proc. 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2006, pp. 357–366.

[26] M. S. Granovetter, The strength of weak ties, *Am. J. Sociol.* **78**(6) (1973) 1360–1380.

[27] R. Burt, *Structural Holes: The Social Structure of Competition* (Harvard University Press, 1995).

[28] N. Lin *et al.*, Social resources and strength of ties: Structural factors in occupational status attainment, *Am. Sociol. Rev.* **46**(4) (1981) 393–405.

[29] P. V. Marsden and K. E. Campbell, Measuring tie strength, *Soc. Forces* **63**(2) (1990) 482–501.

[30] E. Gilbert and K. Karahalios, Predicting tie strength with social media, in *Proc. 27th ACM Conf. Human Factors in Computing Systems*, 2009, pp. 211–220.

[31] Y. Choi and W. Szpankowski, Compression of graphical structures: Fundamental limits, algorithms, and experiments, *IEEE Trans. Inf. Theory* **58**(2) (2012) 620–638.

[32] R. Ahlswede, E.-H. Yang and Z. Zhang, Identification via compressed data, *IEEE Trans. Inf. Theory* **43** (1997) 48–70.

[33] U. N. Raghavan, R. Albert and S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, *Phys. Rev. E* **76** (2007) 036106.

[34] S. Chen and J. Reif, Efficient lossless compression of trees and graphs, in *Proc. IEE Data Compression Conf.*, 1996, pp. 428–437.

[35] A. Gupta and S. Verdu, Nonlinear sparse-graph codes for lossy compression, *IEEE Trans. Inf. Theory* **55** (2009) 1961–1975.

[36] C. Gioran and I. Kontoyiannis, Lossy compression in near-linear time via efficient random codebooks and databases, arXiv:0904.3340 [cs.IT].

[37] R. Venkataramanan, T. Sarkar and S. Tatikonda, Lossy compression via sparse linear regression: Computationally efficient encoding and decoding, arXiv:1212.1707 [cs.IT].

[38] S. Navlakha, R. Rastogi and N. Shrivastava. Graph summarization with bounded error. In *Proc. ACM SIGMOD Int. Conf. Management of Data*, 2008, pp. 419–432.

[39] W. Fan, X. Wang and Y. Wu, Querying big graphs within bounded resources, in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 2014, pp. 301–312.

[40] A. Ingber, T. Courtade and T. Weissman, Quadratic similarity queries on compressed data, in *Proc. Data Compression Conf.*, 2013.

[41] A. Kaur, N. S. Sethi and H. Singh, A review on data compression techniques, *Int. J. Appl. Res. Comput. Sci. Softw. Eng.* **5**(1) (2015) 769–773.

[42] F. Vecchio, Brain network connectivity assessed using graph theory in frontotemporal dementia, *Neurology* **81**(2) (2013) 134–143.

[43] R. Fritsch and G. Fritsch, *The Four-Color Theorem: History, Topological Foundations, and Idea of Proof* (Springer, 1998).

[44] M. R. Garey and D. S. Johnson, Appendix: A list of NP-complete problems, in *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman, 1979), pp. 199–200.

[45] R. L. Graham and P. Hell, On the history of the minimum spanning tree problem, *Ann. Hist. Comput.* **7**(1) (1985) 43–57.

[46] F. S. Roberts and B. Tesman, Optimization problems for graphs and networks, in *Applied Combinatorics*, 2nd edn. (CRC Press, 2009), pp. 640–642.

[47] L. Euler, Solutio problematis ad geometriam situs pertinentis, *Comment. Acad. Sci. U. Petrop.* **8** (1736) 128–140.

[48] R. Shields, Cultural topology: The Seven Bridges of Königsburg, 1736, *Theory Cult. Soc.* **29**(4–5) (2012) 43–57.

[49] M. Bóna, Do not cross: planar graphs, in *A Walk Through Combinatorics: An Introduction to Enumeration and Graph Theory* (World Scientific, 2011), pp. 275–277.

[50] K. L. Hoffman, M. Padberg and G. Rinaldi, Traveling salesman problem, in *Encyclopedia of Operations Research and Management Science* (Springer, Boston, 2013), pp. 1573–1578.

[51] G. B. Dantzig and D. R. Fulkerson, On the max-flow min-cut theorem of networks, Report No. P-826, RAND Corporation, California (1964).