

SURVEY

Path Querying in Graph Databases: A Systematic Mapping Study

ROBERTO GARCÍA^{1,3} **AND RENZO ANGLES**^{2,3}¹Engineering Systems Doctoral Program, Faculty of Engineering, Universidad de Talca, Curico 3340000, Chile²Department of Computer Science, Faculty of Engineering, Universidad de Talca, Curico 3340000, Chile³Millennium Institute for Foundational Research on Data (IMFD), Santiago 7820436, Chile

Corresponding author: Roberto García (roberto.garcia@utalca.cl)

The work of Roberto García was supported by Comisión Nacional de Investigación Científica y Tecnológica (CONICYT) Programa Formación de Capital Humano Avanzado (PFCHA)/Beca de Doctorado Nacional/2019 under Grant 21192157. The work of Renzo Angles was supported by Agencia Nacional de Investigación y Desarrollo (ANID) Fondo Nacional de Desarrollo Científico y Tecnológico (FONDECYT), Chile, under Grant 1221727.

ABSTRACT Path querying refers to the evaluation of path queries in a graph database. New research in this topic is crucial for the development of graph database systems as path queries are associated with relevant use-cases and application domains. The aim of this article is to identify and establish what is currently known about path querying in graph databases. To achieve this, we conducted a systematic mapping study (SMS) in which we explored four digital libraries and collected research papers published from 1970 to 2022. These articles were filtered, classified and analyzed to extract quantitative and qualitative information which is presented in this article. Additionally, we provide a concise description of keywords, use-cases and application domains associated with path querying in graph databases.

INDEX TERMS Path, path querying.

I. INTRODUCTION

In graph theory, the term *Path Querying* refers to the process of searching for specific paths within a graph or hierarchical data structure. Recall that, at its simplest form, a graph is an abstract structure consisting of nodes and edges, such that each edge connects a pair of nodes; and a path is a sequence of nodes and edges connecting a pair of nodes.

In graph data management, Path Querying refers to the evaluation of path queries in a graph database, i.e. a type of database in which the schema and the data are represented as a graph or a generalization of graphs [1]. A path query allows to search paths satisfying a given criteria, involving constraints over labels (types), properties, or combinations thereof [2].

The research around path querying is crucial for the development of graph database systems [3] because path queries are associated with use-cases in various application domains. For instance, in social networks, the shortest paths between two users are used to model search ranking functions [4], to analyze the characteristics of the communities [5], to find influential users [6], or to study the behavior

of information spreading [7]. In banking and financial services, path querying is used for fraud detection [8], [9]. In bioinformatics, path queries are used for alignment of proteins [10], for analysis of cellular networks [11], and to extract meaningful pathways from protein interaction networks [12].

In line with the above, the community of graph data management has emphasized the significance of path querying [3], and existing graph query languages are providing and creating new features associated with paths. In particular, SQL/PGQ [13] (the fragment of the standard ISO SQL:2023 that allows querying graphs on top of relational databases) and GQL [14] (the proposal of standard query language for graph databases), included features like returning paths (instead of tables of solutions), multiple path semantics (to avoid infinite solutions), and grouping of path query solutions.

Although there are several works that review the advancements in querying graph databases (see Table 1), we could not find a specific study about path querying in graph databases.

This article presents a systematic mapping study (SMS) about path querying, with a focus limited to graph databases. In this sense, we identify, categorize, and analyze existing

The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita.

literature relevant to the research topic. Additionally, we present a comprehensive organization and description of key terms and concepts related to path querying (e.g. “path pattern” or “path discovery”). Our goal was to provide a clear definition for each term, with the intention of providing future researchers with easily accessible information on important concepts associated with path querying in graph databases.

The rest of the paper is organized as follow: the related work is presented in Section II; the research method is presented in Section III; the preparation phase is presented in Section III-A; the gathering phase is presented in Section III-B; the analysis phase is presented in Section IV; our review of research topics is presented in Section V; finally, the conclusions are presented in Section VI.

II. RELATED WORK

The review of the related work that is presented in this section was divided into three groups: graph databases, graph queries, and graph query languages. Each group includes articles that provide surveys, reviews, revisions, comparisons (similar to a systematic mapping study), or analysis based on the topic. The articles are presented in chronological order.

A. GRAPH DATABASES

In 2008, Angles and Gutierrez [1] presented a systematic review of graph database models. Their research discusses various data models and query languages that form the foundation of modern graph database systems. While this survey mentions significant articles on path querying, it does not offer an analysis of these articles.

In 2012, Angles [15] presented a comprehensive comparison of different graph database models. This research encompasses an examination of data models, query languages, and their overall characteristics. It also entails a general analysis of fundamental path queries that were supported by the available query languages at that particular time.

In 2015, Wang et al. [16] published a review of graph database systems and their applications. The study provides general information regarding query languages and query processing techniques used by the analyzed systems.

In 2021, Rabuzin and Šestak [17] presented a comprehensive overview of graph database systems, discussing their history, current state, and potential developments. The article includes an analysis of trends in this field and offers recommendations for future improvements to such systems.

In 2021, Timón-Reina et al. [18] conducted a literature review to investigate the development, efficiency, and utilization of graph database systems. The authors have a goal of applying graph-oriented technologies in the field of bio-medicine.

B. GRAPH QUERIES

In 2011, Barcelo et al. [19] analyzed the foundations of querying graph patterns. The authors identify the key features

of patterns (i.e. a sub-graph with variables and regular expressions). They investigated standard graph queries and provided exact descriptions of data and combined complexity for various pattern classes.

In 2013, Bagan et al. [20] investigated the evaluation of regular path queries, and studied additional restrictions. Furthermore, they proposed a classification scheme for regular languages according to their complexity.

In 2018, Bonifati and Dumbrava [21] conducted a study on various types of graph queries. They analyzed query languages that offer a well-balanced combination of theoretical principles and practical applications. The article also provides a review of diverse query evaluation techniques, encompassing the concepts of “Query approximation” and “Query learning”.

In 2021, Wang et al. [22] published a survey about typical graph queries with attributes. They developed a taxonomy based on the inputs and outputs of these queries. Their research encompasses a collection of various query types organized by domain, and for each query, they provide an examination of its meaning and the algorithms used to resolve it.

C. GRAPH QUERY LANGUAGES

In 2012, Wood [23] presented a brief revision of graph query languages and discussed the core functionalities provided by these languages, including subgraph matching, finding nodes connected by paths, comparing and returning paths, aggregation, node creation, and approximate matching and ranking. The author also review theoretical results related to expressive power and complexity of evaluating graph queries.

In 2017, Angles et al. [24] presented a revision of graph data models and practical graph query languages. The primary focus of this article was on graph pattern queries and navigational queries (also known as path queries), along with a detailed discussion on various semantics and complexity results. Additionally, the article provided several examples to highlight the syntactic variations between different query languages commonly used in this field.

In 2018, Angles et al. [2] defined the concept “graph query language” as the way to retrieve or extract data which have been modeled as a graph and whose structure is defined by a graph data model. This article presents a summary of the main types of regular path queries, along with their associated challenges and key applications.

In 2021, Sharma et al. [25] published the results of a systematic literature review on graph query languages. The purpose of this article is to review the main features, current methods, and the types of graph patterns supported by the existing query languages.

After reviewing the literature, we have found multiple articles that discuss graph databases, graph queries and graph query languages, including systematic reviews and surveys. However, we did not find any research specifically focused on path querying. Therefore, conducting a systematic mapping study on path querying in graph databases would

TABLE 1. Comparison with related works.

Ref.	Summary	Year	Type of work	Focus			
				Graph databases	Graph queries	Graph query languages	Path querying
[1]	Systematic review of graph database models	2008	Review	✓	✗	✗	✗
[15]	Graph data models comparison	2012	Comparison	✓	✗	✗	✗
[16]	Review of graph database systems and applications	2015	Review	✓	✗	✗	✗
[17]	Review of graph database systems	2021	Review	✓	✗	✗	✗
[18]	Review the development, efficiency, and utilization of graph database systems	2021	Review	✓	✗	✗	✗
[19]	Analysis the foundations of querying graph patterns	2011	Analysis	✗	✓	✗	✗
[20]	RPQ evaluations	2013	Analysis	✗	✓	✓	✗
[21]	Study of graph queries and review of evaluations techniques	2018	Analysis	✗	✓	✗	✗
[22]	Survey about typical graph queries with attributes	2021	Survey	✗	✓	✗	✗
[23]	Presented a brief revision of graph query languages and discussed the core functionalities provided by these languages	2012	Review	✗	✗	✓	✗
[24]	Revision of graph data models and practical graph query languages	2017	Review	✗	✗	✓	✗
[2]	This article presents a summary of the main types of graph queries	2018	Survey	✗	✗	✓	✗
[25]	SLR on graph query languages	2021	SLR	✗	✗	✓	✗
This work	SMS on path querying in graph databases	2023	SMS	✓	✗	✗	✓

be a valuable starting point for a comprehensive review and detailed analysis of this field. Additionally, the findings presented in this article highlight unexplored features and could inspire the creation of new query languages.

III. RESEARCH METHOD

The development of this article is based on the notion of Systematic Mapping Study (SMS), which is a research method that involves an extensive review of primary studies in a specific topic area with the objective of identifying the available evidence on the topic [26]. The process entails conducting a literature search to determine which topics have been previously addressed and where the research has been published [27]. The result of a systematic mapping study is a collection of papers related to the topic area, organized according to a classification. Therefore, a mapping study offers an overview of the range of the subject area and enables the identification of research gaps and trends [27].

Based on the proposal made by Petersen et al. [28], our systematic mapping study follows a process comprising three phases:

- *Preparation phase*, where several parameters and criteria to guide the systematic mapping are defined. In essence, research objectives, research questions, bibliographical sources, search terms, and selection and relevance criteria are defined in this phase.
- *Gathering phase*, which is oriented towards gathering relevant research papers according to the parameters and criteria defined in the preparation phase. In essence, we search, gather, and select candidate works. Then, we classify the works and select the relevant ones based on their classification. Finally, we gather new search terms and repeat this phase with the new terms.

- *Analysis phase*, that involves extracting information and knowledge from the research papers obtained during the gathering phase. The research objectives and research questions defined in the preparation phase guide this phase. The result of this phase is a report that includes both quantitative and qualitative analyses.

In the following sections, we will present the results of applying the phases described above.

A. PREPARATION PHASE

In this section, we will outline the research objectives, research questions, bibliographical sources, search terms, general selection criteria, and relevance criteria.

1) RESEARCH OBJECTIVES

The main goal of the systematic mapping described in this article is to gather, categorize, and analyze the majority of research studies that are concerned with “path querying” in the field of graph data management. The specific aims of this project are:

- Provide statistical information about the research papers related to path querying in graph databases;
- Identify topics and research lines in the area of path querying, and raise awareness about their level of development;
- Identify application domains and uses cases where path querying is applied.

2) DEFINITION OF RESEARCH QUESTIONS

The systematic mapping presented in this article has been designed to address the following research question:

- RQ1. What terms associated with “path querying” in graph databases are the most influential?

This question is oriented to know the most important terms associated to the notions of “path querying” in graph databases. Such terms will be collected from the articles found in the bibliography, and those with the highest frequency of use will be considered influential.

RQ2. What kinds of research papers have been developed in the context of “path querying” in graph databases?

This question will deliver information about the types of research papers found in the literature. We consider two classifications: based on deepness (define, implement, optimize, learn, or compare) and based on scope (use case, practical, survey or theoretical).

RQ3. How the research related to “path querying” in graph databases has evolved?

This question is oriented to know the evolution of the associated terms in the time. Specifically, we would like to know the periods of time with more and less activity for each associated term. It is also useful to rank the associated terms according to their level of activity.

RQ4. Where the research papers related to “path querying” in graph databases are published?

This question aims to find out the scientific events (conferences and workshops) and journals in which research on path querying in graph databases has been published. It is helpful to know where to publish new research related to both topics.

RQ5. What query languages allow “path querying”?

This question aims to discover which query languages support operations for path querying. We are interested in specific aspects such as language syntax, language semantics, expressiveness, and computational complexity.

RQ6. What are the application domains for “path querying”?

We would like to gather information about the application domains and the specific use-cases related to path querying in graph databases. We desire to understand the challenges that have been addressed and resolved by utilizing path queries and graph databases.

3) BIBLIOGRAPHICAL SOURCES

The search for research papers will be conducted on the following databases of bibliographic literature: ACM Digital Library [29], IEEE Xplore [30], Springer [31], and Scopus (Elsevier) [32].

4) SEARCH TERMS

In order to define the search terms, we conducted an initial exploration of the bibliographic sources by using the keyword “Path”. As shown in Table 2, this search returned a high number of articles, in particular for Springer. After

TABLE 2. Research papers found in each bibliographic source, using only the “Path” keyword.

Bibliographic Source	Results
ACM Digital Library	185,997
IEEE Xplore	181,720
Springer	1,147,479
Scopus	899,056

quickly reading the returned articles, which included the title, abstract, and keywords list, we defined the following list of initial search terms: Path querying, Path manipulation, Path query, Path query language, Path operation, Path expression, Path algebra, Path pattern, Path evaluation, Path discovery, Path finding, and Path retrieval.

Additionally, we discarded research papers related to the following terms: Computer path (files), Management operations, Network analysis, Optimization, Path constraint, Path database, Path data model, Path language, Path logic, Path syntax, and Printers.

5) GENERAL SELECTION CRITERIA

A research paper is considered a “candidate paper” if it satisfies the following conditions:

- Publication year: Higher than 1970;
- Language: English;
- Article type: Journal, conference or workshop;
- Knowledge areas: Computer Science, Data Management, Graph Databases.

6) RELEVANCE CRITERIA

We considered four levels of relevance:

- *Highly Relevant*: A research paper that proposes, develops or compares fundamental elements related to path querying in graph databases, including data models, query languages, algorithms, methods or strategies.
- *Moderately Relevant*: A research paper that compares models and methods related to path querying in graph databases, or describes their application in specific application domains.
- *Somewhat Relevant*: A research paper that is not related to path querying in graph databases, however the models and methods presented in the paper can be a source of inspiration for the area of path querying.
- *Not Relevant*: A research paper that corresponds to the search terms, but it is not related to the field of data management.

A paper will be considered a “Relevant Work” if it fits into one of the top two levels defined above.

B. GATHERING PHASE

This section presents the results of the gathering phase of our systematic mapping. It includes the search of research papers, the selection of candidate works, the classification of papers, the selection of relevant papers, and the review of search terms.

1) SEARCH AND GATHERING OF WORKS

The search of papers depends on the search engine provided by each bibliographic source. Next we present the search string used in each bibliographic source.

ACM digital library:

```
Title:("KEYWORD") OR
Abstract:("KEYWORD")OR
Keyword:("KEYWORD")
```

IEEE Xplore:

```
"Document Title": "KEYWORD" OR
"Abstract": "KEYWORD" OR
"Index Terms": "KEYWORD"
```

Springer:

```
"KEYWORD"
```

Scopus:

```
(TITLE-ABS-KEY ("KEYWORD" )) AND
(LIMIT-TO( LANGUAGE , "English")) AND
(LIMIT-TO (DOCTYPE , "cp") OR
LIMIT-TO (DOCTYPE , "cr" ) OR
LIMIT-TO (DOCTYPE , "ar" ) ) AND
(LIMIT-TO (SUBJAREA , "COMP")) AND
(LIMIT-TO (SRCTYPE , "p") OR
LIMIT-TO (SRCTYPE , "j" ))
```

Please take note that each bibliographic source employs a distinct search string, and there are limitations on the available filtering options in certain cases. Therefore, it was imperative to utilize the advanced search functionalities to fulfill all the selection criteria, such as the language requirement.

2) SELECTION OF CANDIDATE WORKS

The search of the bibliographic sources with the selected keywords, yielded a total of 9193 works. To ensure accurate search results, we checked that each papers included the appropriate search term in its title, abstract, or list of keywords, performing a screening of these articles and discarding those that did not have a direct relationship with “Path Querying” and “Graph Databases”, like articles related with mathematics (path algorithms), operation management (path optimization), internet networks (shortest paths), among others.

In Table 3, we show the number of candidate research papers discovered in each bibliographic source, totaling 305, where 67 papers are shared among the sources, totaling 238 unique papers. While Scopus provided the largest number of papers, some of them are also accessible in the other bibliographic sources. Furthermore, the list of research papers was supplemented with two papers suggested by the authors of this paper.

Additionally, in Table 4 we show the number of works for each search term and bibliographic source. Initially, it is evident that the term “Path query” has the highest number of results, particularly in Scopus. Conversely, the term “Path manipulation” did not yield any results.

TABLE 3. Candidate research papers found in each bibliographic source. A total of 238 unique papers were found (where 67 articles are shared among the sources). The column “Repeated” indicates the number of results shared with other digital libraries.

Bibliographic Source	Results	Repeated
ACM Digital Library	62	2
IEEE Explore	62	1
Springer	49	6
Scopus	130	58
Suggested	2	0

TABLE 4. Number of research papers organized by search term and bibliographic source.

Search Term	IEEE Xplore	ACM DL	Springer	Scopus
Path querying	5	6	3	17
Path manipulation	0	0	0	0
Path query	21	25	20	41
Path query language	1	3	3	12
Path operation	0	0	0	1
Path expression	11	12	8	21
Path algebra	4	2	4	8
Path pattern	2	3	3	7
Path evaluation	0	1	0	3
Path discovery	2	3	4	15
Path finding	16	4	4	4
Path retrieval	0	3	0	1

3) CLASSIFICATION OF CANDIDATE WORKS

In order to provide a general comparison of the candidate papers, we defined the following three classifications:

- Keyword-based classification: Each candidate papers was related to one or more search terms listed in Section III-A4.
- Work type classification:
 - Theoretical: An article that presents a formal development of the concepts, theories and methods related to a research topic.
 - Practical: An article that describes an specific model or method which can be applied to an abstract or real problem.
 - Use case: An article that shows the use of a model or method in an specific application domain.
 - Survey: An article that compares the models and methods related to a research topic.
- Work focus classification:
 - Definition: An article that defines a model or method for path querying in graph databases.
 - Implementation: An article that describes the implementation of a model or method for path querying in graph databases.
 - Optimization: An article that describes optimization methods for path querying in graph databases.
 - Learning: An article that applies methods based on automatic learning and prediction for path querying in graph databases.

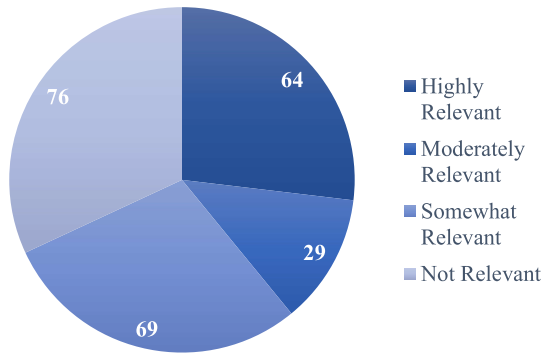


FIGURE 1. Relevance of candidate research papers.

- Comparison: An article that compares operations, languages and methods associated with path querying in graph databases.

These classifications will be used later (Section IV) in order to address the research questions.

4) SELECTION OF RELEVANT WORKS

In this step, each candidate paper was reviewed in order to determine its relevance according to the criteria defined in Section III-A6. Figure 1 shows that, for a total of 238 articles, 64 (27%) were considered Highly Relevant, 29 (12%) as Moderately Relevant, 69 (29%) as Somewhat Relevant, and 76 (32%) as Not Relevant. Based on the inclusion criteria defined in Section III-A6, 93 (39%) papers were selected for the next step (Highly Relevant and Moderately Relevant), while 145 (62%) were excluded (Somewhat Relevant and Not Relevant).

Please note that the chosen papers in this classification represent less than half of the papers chosen in the previous stage. This indicates that the search terms are broad and apply to multiple knowledge areas, resulting in a significant number of unrelated papers during the search process.

5) REVIEW OF SEARCH TERMS

The relevant papers were reviewed in order to find new search terms. To do this, for each article, we read the “Abstract”, “Keywords” and “Introduction” sections.

This step resulted in six additional search terms: “Finding paths”, “Navigational Query Language”, “Path properties”, “Path query evaluation”, “Path sequence” and “Path traversal”. According to our methodology, the gathering phase was repeated by using the above search terms. After the search step we obtained 1759 works, of which 18 works were selected as candidates, and 4 were selected as relevant.

The second examination of search terms yielded no additional search terms, therefore the total number of papers acquired during the Gathering Phase was 97. This total includes 93 articles obtained in round 1, plus 4 articles obtained in round 2.

TABLE 5. Terms related to path querying in graph databases. The table also indicates the number of found papers and the number of selected papers for each term.

Term	Found papers	Selected papers
Path query	90	44
Path expression	43	14
Path discovery	22	11
Path query language	13	7
Path algebra	15	7
Path querying	18	6
Path pattern	14	5
Path finding	28	4
Path evaluation	4	1
Path operations	1	1
Finding paths	5	1
Navigational query language	2	1
Path properties	2	1
Path traversals	4	1
Path retrieval	4	0
Path query evaluation	3	0
Path sequence	2	0
Path manipulation	0	0

IV. ANALYSIS PHASE

In this section, we will address the research questions that were proposed during the Preparation phase. The answers provided here are derived from the 97 research papers that were selected during the gathering phase.

A. WHAT TERMS ASSOCIATED WITH PATH QUERYING IN GRAPH DATABASES ARE THE MOST INFLUENTIAL? (RQ1)

The terms associated with path querying in graph databases, which were obtained during the Gathering phase, are listed in Table 5. It is evident that the term “Path Query” has the highest number of selected papers, while the term “Path manipulation” is not mentioned in the literature. It should be noted that the number of selected papers is less than half of the total number of papers found. It is worth recalling that some papers were excluded during the selection phase as they were focused on different research areas such as database optimization, semi-structured databases, and specific application domains.

B. WHAT KINDS OF RESEARCH PAPERS HAVE BEEN DEVELOPED IN THE CONTEXT OF “PATH QUERYING” IN GRAPH DATABASES? (RQ2)

In order to answer this question, we will use the “Paper Type” and “Paper Focus” classifications used during the gathering phase. In the first case, a research paper can be classified as Theoretical, Practical, Use-case or Survey. In the second case, we have five classes: Definition, Implementation, Optimization, Learning and Comparison. The number of papers in each classification is shown in Figure 2. The specific references for both categories are compiled in Table 7 and Table 8.

The papers that are labeled as “Theoretical” propose various methods for representing, storing, and querying paths.

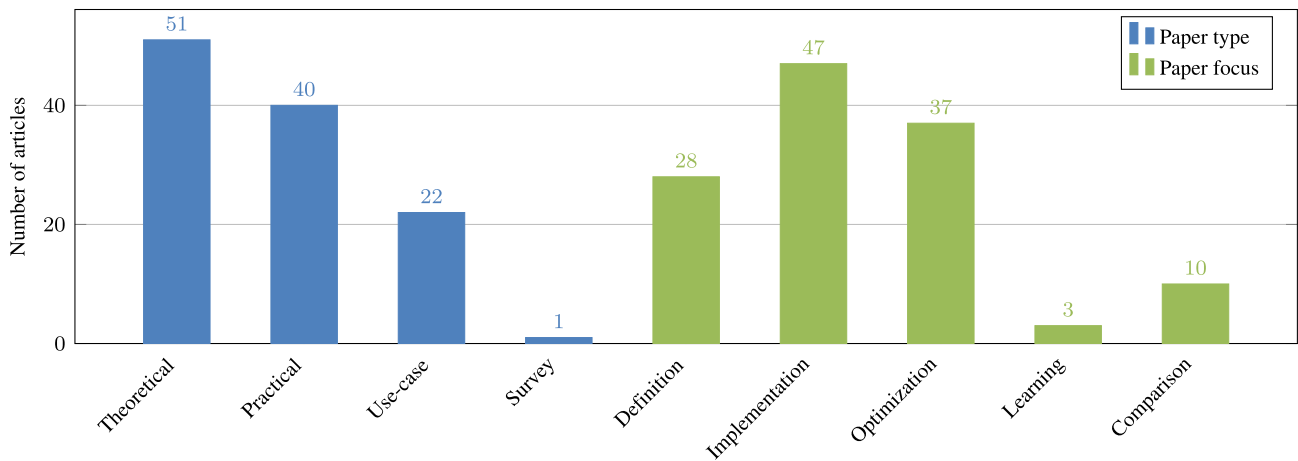


FIGURE 2. Research papers classified by type and focus.

These methods also include analysis of their expressiveness and computational complexity. On the other hand, the papers categorized as “Practical” describe the implementation of methods and algorithms used for storing and querying paths. These papers also provide empirical evaluations of their effectiveness. The class “Use-case” includes papers describing the use of paths in different application domains, including workflow provenance [33], road networks [34], [35] and communication networks [36]. In [22], the single paper classified as “Survey”, the authors propose a taxonomy of attributed graph queries, including an analysis of semantics and algorithmic motivations behind such queries. In some cases, a work can be classified in more than one category. For example, Bai et al. [37] present theoretical and practical results related to the optimization of path queries.

Regarding the classification based on the focus of the paper, most of the research is centered around implementing methods for path querying, followed by optimization techniques. The use of machine learning techniques to enhance the evaluation of path queries is a relatively new research area.

C. HOW THE RESEARCH RELATED TO “PATH QUERYING” IN GRAPH DATABASES HAS EVOLVED? (RQ3)

In this case, we analyzed the popularity of the research terms (associated with the relevant papers) in different periods of time, as shown in Figure 3. Starting in 1970, we can see a significant occurrence of terms like “path expression” and “path algebra”, with a quiet period at the end of the eighties. The term “path query” began to gain popularity during the nineties, and it remained relevant between 2010 and 2015, and it is still popular to this day. The term “path expression” also gained popularity in the nineties due to the research on semi-structured databases and the development of XML technologies.

From 2010 to 2020, there was a significant emphasis on “path query languages”. Specifically, researchers concentrated on studying regular path queries (RPQs), which serve

as the fundamental method for expressing path queries [38]. The term “path discovery” also gained popularity during this period due to its association with the utilization of machine learning techniques in identifying network connections.

In the last years, the three most popular terms are “path querying”, “path discovery” and “path pattern”. The last term has gained interest because it poses a challenge for future graph query languages.

D. WHERE THE RESEARCH PAPERS RELATED TO “PATH QUERYING” IN GRAPH DATABASES ARE PUBLISHED? (RQ4)

The venues and journals where the research papers related to path querying in graph databases have been published are shown in Figure 4 (the name associated to each acronym is given in Appendix 9). We discovered papers in significant conferences such as the International Conference on Data Engineering (ICDE), the Symposium on Principles of Database Systems (PODS), and the International Conference on Management of Data (SIGMOD).

The Figure 5 shows the distribution of the research paper according to the type of publication venue. We found that most of the papers were published in conferences, followed by journals, and a small fraction in workshops. This indicates that the field is dynamic and evolving, as conferences are more suitable for presenting new ideas and receiving feedback.

E. WHAT GRAPH QUERY LANGUAGES ALLOW “PATH QUERYING”? (RQ5)

The notion of path querying has been considered in several query languages, and the specific features depend on the data management approach and the corresponding data model. In particular, we can identify graph query languages like G-CORE [39] and Cypher [40], Semantic Web oriented query languages like SPARQL [41], and Web oriented query languages like GraphQL [42]. The specific path-oriented

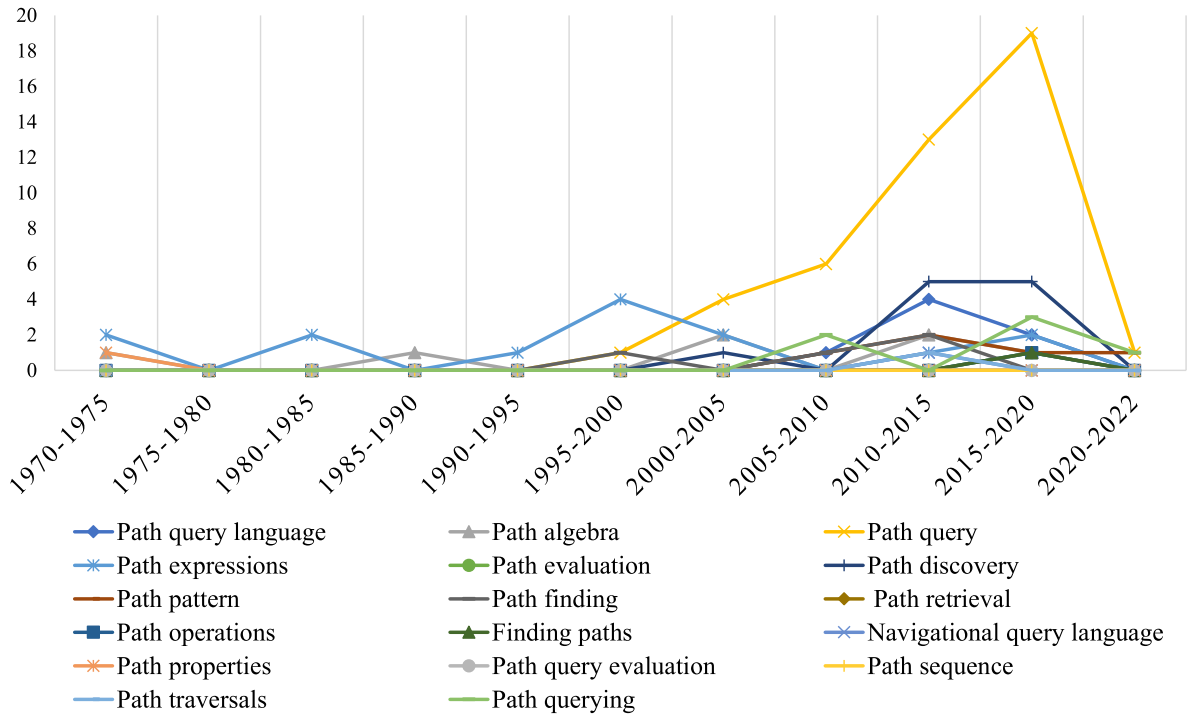


FIGURE 3. Popularity of search terms by year of publication.

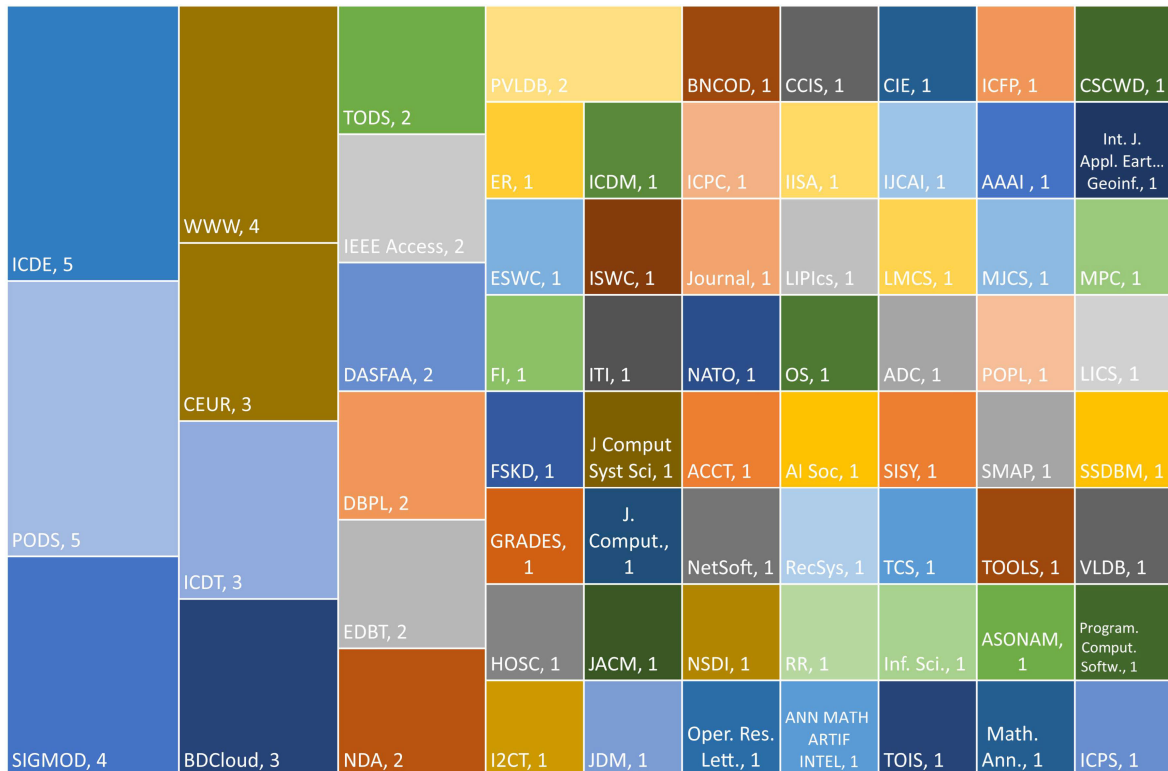


FIGURE 4. Journals and conferences where articles related to path querying in graph databases have been published.

features supported by theoretical and practical graph query languages can be reviewed in papers like [24] and [43].

The functionalities for querying paths and their subsequent manipulation are just supported by two graph query

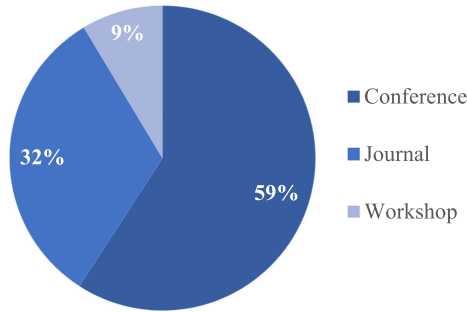


FIGURE 5. Research papers distribution by venue.

languages: G-CORE and Cypher. G-CORE defines a data model where paths are first class citizens (i.e. paths can be created and stored), the query language supports a powerful and tractable set of regular path queries, but the functions for path creation are very simple. Cypher [40] supports some types of path queries (e.g. shortest path, fixed and variable length path), and the resulting paths are represented as a sequence of nodes and edges, in such a way that paths can be manipulated by using list-oriented functions (e.g. create, combine and divide).

F. WHAT ARE THE APPLICATION DOMAINS FOR "PATH QUERYING"? (RQ6)

We found the following application domains: Criminology [44], News extraction [45], Work Flow [33], Genealogy [46], Geography [47], Conceptual maps [48], Communication networks [36], [49], [50], Social networks [22], [51], [52], [53], Road networks [34], [35], [54], [55], [56], [57], [58], Process synchronization [59], and Communication networks [50]. The number of articles related to each application domain is shown in Figure 6. We can see that path queries are mostly used in Communication Networks, Social networks and Road networks.

A Social Network is represented as a graph where the nodes represent people and the edges represent different types of relationships (e.g. familiar and friendship). Path operations are used to determine relevant people (e.g. influencers or bridges), relevant groups (clustering), possible connections, and distance between people.

In a graph representing a road network, each node represents an intersection (e.g. a city), and each edge (u, v) represents a road segment that enables traversal from u to v. A common use-case, that implies path operations, is to find the optimal route between places, considering distance and expenses efficiency.

V. REVIEW OF RESEARCH TOPICS

Based on the search terms used in the previous phases, we defined a collection of research topics grouped in four categories (see Table 6): General terms, Path representation, Path querying and Application domain (with use cases). For each research topic, we presented a short description based on

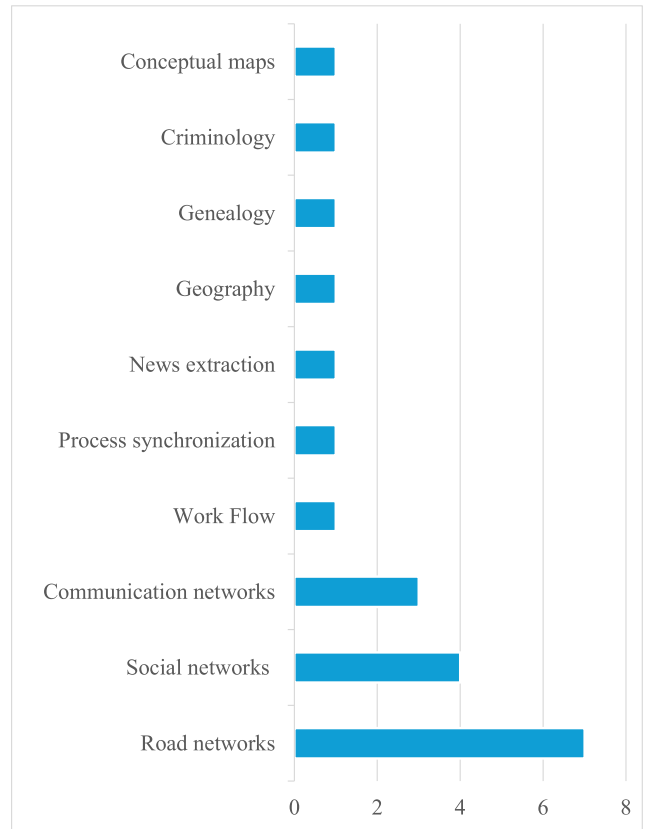


FIGURE 6. Number of papers in each application domain.

TABLE 6. Research topics (grouped in four categories) associated with path querying and path manipulation.

Category	Research topics
General	Path problem (shortest path problem), path retrieval, path evaluation, path finding, path planning, path expression query processing, path query evaluation, path discovery, shortest path discovery.
Path representation	Path expression, path pattern, path sequence, path traversal, property path, regular path pattern, short path, shortest path.
Path querying	Path query, general path expression query, navigational query, path pattern query, property path query, reachability query, regular path query (regular simple path, context-free path query, conjunctive regular path query), shortest path query, path query language, navigational query language, path manipulation, path algebra, path operation.
Application domain	Communication network, finding path, road network (route query, taxi trajectory), semantic network, social network (social network analysis), software life cycle activities, routing (route searching, traffic routing).

the relevant papers obtained during the gathering phase. The aim of this section is to offer easily accessible information regarding important concepts associated with path querying in graph databases.

A. GENERAL TOPIC

1) PATH PROBLEM

The term “path problem” can be seen as a unified framework for problems found in different fields [60]. In graph theory, the most studied path problem is the shortest path problem [61]. In the context of data management, a typical path problem pertains to effective resource management during the computation of the shortest path. This involves minimizing the amount of the path stored in memory, facilitating reduced memory usage and the utilization of dynamic paths [62]. Additionally, path problems can also be associated with a range of enumeration and optimization problems that focus on generating or comparing paths in graphs [63].

2) SHORTEST PATH PROBLEM

Consider a graph in which each edge is connected to a weight or cost. When given a pair of nodes (s, t), the shortest path from s to t is a path that meets the condition that no other path has a lower weight [64]. This is a highly researched problem in graph theory and has two well-known algorithms for its solution, namely the *Bellman-Ford* and *Dijkstra's* [61].

3) PATH RETRIEVAL

Given a graph G , and a query Q that searches for a path between two nodes represented as P , the term “path retrieval” refers to the process of finding a set of paths that satisfy the query Q on G [65]. Path retrieval enables users to determine if a path exists between a pair of nodes, and if so, it returns the paths that connect those nodes [66]. The term path retrieval is utilized in various applications domains. For example, in transportation networks, path retrieval refers to the algorithms used to retrieve optimal paths [67].

4) PATH EVALUATION

Defined as the evaluation of a path with the purpose of obtaining its cost, which can be its length (the number of edges between them) or some other property. Some of the properties that can be assessed for a path include normalization, monotonicity, relocation, homogeneity, and additivity [68].

5) PATH FINDING

This term is used to describe the process of finding a path between any two nodes in a graph. The result depends on the algebra used and can be either a single path or a set of paths. To explore this concept, Manger [63] investigated various algebras for finding paths in graphs. Depending on the selected algebra, it can return all possible paths, only simple paths (without repeated nodes), or only elementary paths (without repeated edge labels).

6) PATH PLANNING

This concept is usually utilized in the context of navigation devices, localization based services (GPS), and autonomous vehicles [34]. Seen as a graph, the nodes represents points of

interest, and the edges represent the roads that connects the points. This graph has properties, such as time or distance, and the objective is to obtain the optimal path based on minimizing one of such properties [35]. In a traffic path planning of vehicle navigation, the selection of a path with minimum traveling time is mostly the major objective [35]. The most common used algorithms are the Dijkstra algorithm and the A-star algorithm [35].

7) PATH EXPRESSION QUERY PROCESSING

It means the application of processing techniques for computing path queries. Among the available methods are: *Forward traversal*, for path expressions that involve a selection operation on the start of the traversal, and *Reverse traversal*, for path expressions that involve access to the end of the path traversal [69].

8) PATH QUERY EVALUATION

It concerns the methods and techniques used to evaluate path queries. There are two approaches: the first approach is called *direct evaluation*, where the path is traversed at the query time exploring algorithms like A* or depth-first search [70]; the second approach is called *pre-processing the graph*, meaning that the graph is pre-processed in order to obtain reachability information that can be used to make a more efficient evaluation, but requiring more disk space [71].

9) PATH DISCOVERY

This terms is equivalent to “path querying” Defined as the search of one or more paths between a pair of nodes [72]. According to Hong et al. [73], it is one of the most important search queries for large dynamic graphs representing complex networks, as social networks, website networks, telephonic networks, among others. Usually the most used algorithms for their resolution are related to the search of the shortest path.

10) SHORTEST PATH DISCOVERY

This is a generalization of the shortest path problem (see above). Given a directed edge-weighted graph, this term concerns the task of finding the shortest path for fixed source and target nodes such that initially the edge-weights are unknown, but they can be queried. Querying the cost of a path is expensive and hence the goal is to minimize the total number of edge cost queries executed [72].

B. PATH REPRESENTATION

1) PATH EXPRESSION

In the most general sense, a path expression is a way to specify a path from a source element to a target element in a database. For example, in the relational model, a path expression specifies a path from a source relation to a target relation rather than using some type of join operation [46]. In semi-structured query languages, a path expression represents a sequence of steps from a source node

(e.g. the root node of the data tree) [74] to a target node in the data tree. In the context of graph query languages, a path expression represents a path from a source node to a target node in a data graph [75]. A path expression is also called a regular path expression.

2) PATH PATTERN

A path pattern is a path extended with conditions on its nodes, edges and properties. Such conditions are commonly represented using regular expressions. There are several semantics for evaluating path patterns, including arbitrary path semantics, shortest path semantics, no-repeated-node semantics, and no-repeated-edge semantics [24]. For instance, an evaluation under no-repeated-node semantics returns the set of all paths (also called simple paths) where each node appears once [52], [76].

3) PATH SEQUENCE

Given a graph G and a distinguished source node s , we use $P(s, v)$ to denote the existence of a regular expression P which allows to obtain all paths from s to any other node v in G . A path sequence is a sequence of expressions $(P_1, v_1, w_1), (P_2, v_2, w_2), \dots, (P_l, v_l, w_l)$ such that P_i is an unambiguous path expression of type (v_i, w_i) . A path sequence can be used to solve the single-source path expression problem for any source [77].

4) PATH TRAVERSAL

It is defined as the route between different pair of nodes on a graph by traversing the corresponding edges. In other context, it is recognized as one of the strengths of object oriented query processing, given that it allows the recovery of information through the use of concepts related with the pointer navigation [69].

5) PROPERTY PATH

This term is introduced in SPARQL 1.1, the second version of the standard query language for RDF databases. A property path is a triple pattern (i.e. a tuple of subject, predicate and object) where the predicate is a regular expression, representing an arbitrary length path [78]. A property path allows to express different types of navigational queries [79], [80]. In [81], Losemann and Martens study the complexity of regular expressions and property paths in SPARQL.

6) REGULAR PATH PATTERN

Defined as a path pattern expressed through the use of regular expressions. A regular path pattern allows to express various repetitive patterns and restrictions on the same query [52].

7) SHORTEST PATH

Given a graph G , the shortest path between a pair of nodes x and y is the path with the smallest number of edges (i.e. the length of the path). The term “Short path” is used with the same meaning [82].

C. PATH QUERYING

1) PATH QUERY

It is a fundamental graph query paradigm, which aims to test for the existence of a path between pairs of nodes, and also obtain such paths to return from queries [83]. Path queries consist of finding all nodes that can be reached by traversing a path whose labels form a word in a regular expression over an alphabet of labels [84]. Hellings [85] states that a path query q is specified by the use of a language L which contains all the traces of the path of interest.

2) GENERAL PATH EXPRESSION QUERY

A general path expression query is a type of query that is commonly used in object-oriented database management systems (OODBMS) [86], and consists of more than two classes of node objects connected through a relation. This kind of queries can be made upon multiple two-class path expressions. According to Taniar [69], forward and reverse traversals can be applied to a general path expression query, and an important point to improve is the optimization of this type of query, allowing an indeterminate number of classes to be connected.

3) NAVIGATIONAL QUERY

Navigational queries correspond to property path queries that use recursive operators as Kleene star (*) or Kleene plus (+) [87]. These queries are well known in the context of graph databases [88], [89]. One of the most basic form to include navigation in graph queries is to start with a basic graph pattern and augment it with navigational primitives like regular expressions [90]. Navigational queries are one of the key differences between graph and relational databases. Relational databases are not designed to handle this type of query, whereas graph databases are specifically designed to efficiently handle them. Additionally, there has been extensive paper on query languages that can express complex navigational patterns [91].

4) PATH PATTERN QUERY

A path pattern is a directed labeled path with conditions which represent regular expressions. The result of a path pattern query is a set of all simple paths (i.e. path queries that allow traversing each node only once [20]) satisfying given conditions. Also known as *path pattern matching*, this notion is very important in the management and mining of data with graph structures. Today, many emergent algorithms depends directly or indirectly on the effective computation of paths between two nodes. There are three problems to be solved for the resolution of path pattern queries: how to express clearly and concisely a path query with complex constraints; how to deal with large scale data; and how to make full use of the centric computing platforms [37], [52], [76].

5) PROPERTY PATH QUERY

It is related to search paths in RDF graphs. This is achieved through the introduction of property paths operators (see

Property Path). This queries tend to inevitably coincide with numerous results [88].

6) REACHABILITY QUERY

Given a pair of nodes (a , b), where a is the start node and b is the end node, a reachability query looks for the existence of a path from a to b . If such path exists, then we can say that b is reachable from a . A reachability query is considered as a fundamental query in graphs [92], [93], [94].

7) REGULAR PATH QUERY

A regular path query (RPQ) returns a set of pairs of nodes connected in the database through a path conforming to a regular expression [95]. Formally, an RPQ is defined as an expression $x \xrightarrow{L} y$ where L is a regular language from a labeled graph, and x , y are the start and the end node respectively. The result of an RPQ is a path that complies with the expression L [96]. RPQs are a fundamental to express path queries in graph query languages [97].

8) REGULAR SIMPLE PATH

A regular simple path query (RSPQ) is similar to a regular path query (RPQ), but it allows to go through the same node multiple times, i.e. an RSPQ passes one time on each node. The use of this type of query is desirable in scenarios such as transportation networks, metabolic networks, DNA matching and wireless networks [20].

9) CONTEXT-FREE PATH QUERIES

Context-free path queries (CFPs) are defined as an extension of regular path queries (RPQs), where context-free grammars can be used as constraints in navigational graph queries. CFPs are very powerful and more expressive than traditional RPQs, but their computation in large graphs requires high processing resources [98]. CFQs are used in model verification, bioinformatics and parser's construction [85].

10) CONJUNCTIVE REGULAR PATH QUERIES

A conjunctive regular path query (CRPQ) expresses the existence of patterns in a graph database, which satisfies a series of regular restrictions [91]. CRPQs are not expressive enough for many natural query tasks, and need to be extended for increase their expressive power [99]. CRPQs cannot output paths and they cannot express relation between paths [75].

11) SHORTEST PATH QUERY

Defined as path queries that use shortest path semantics for its resolution, that is, paths of minimal length that satisfy an imposed constraint over the query. Current graph query languages allow to return a single shortest path or all shortest paths [24]. There are several algorithms for computing shortest path queries; one of the most used is *Dijkstra's* algorithm as it performs well for a variety of cases [100].

12) PATH QUERY LANGUAGE

It is a language that allows expression of path queries. Typically, a path query is achieved through the use of regular expressions [101], [102]. A path query language has various properties, such as compositionality, which means that the output of a path query can be used as the input in another query [36].

13) NAVIGATIONAL QUERY LANGUAGE

It is a language used to navigate graph structures, i.e. it includes operations to move from one node to an adjacent one in a graph, recursively when necessary. A navigational query language is usually based on binary relational algebras (like *XPath*) or in regular expressions (like RPQs) [80]. Some navigational query languages have a common disadvantage: they are very suitable for expressing relevant properties from the underlying topology of a graph database, but not about how it interacts with the data. Two languages (*walk logic* and *regular expressions with memory*) have been proposed to overcome this problem [89].

14) PATH ALGEBRA

A path algebra concerns a set of operators and rules which allow the manipulation of paths [103], [104]. Manger [63] defines a path algebra equipped with two binary operations, *join* and *product*. Such algebra can be used to find paths in a graph, specifically, to identify one path between any pair of nodes.

15) PATH MANIPULATION

The term "path manipulation" is not defined in the literature. In the area of data management, the term data manipulation is related not only to the action of querying a database, but also to modify and update data in a database [105]. Hence, we define "path manipulation" as the operations to query, add (insert), delete and modify (update) paths in a database.

16) PATH OPERATION

Informally, a path operation can be defined as an action that can be applied over a path or collection of paths. Examples of path operations are adding a path to a graph, delete a path from a graph and join two paths [106]. Current graph database systems are studying and implementing basic graph operations. For example, Cypher (the query language of Neo4J [40]) provides functions to create a path from a set of elements, combine two paths, create sub-paths from a path, and return the elements of a path.

D. APPLICATION DOMAINS AND USE CASES

1) COMMUNICATION NETWORK

A communication network is comprised of a group of hosts (machines or devices) that provide services to the users, as well as a subnetwork that connects the hosts, enabling communication between them (users and hosts). Networks of this type typically exhibit characteristics of being extensive,

ever-changing, and intricate. A communication network can be modeled as a graph in which machines are depicted as nodes, and each pair of connected machines is represented by an edge [36], [49].

2) FINDING PATH

The concept relates to finding the best path between two nodes. Usually, it consists of reliable path-finding algorithms. Path finding is used in multiple areas, such as GPS navigation, auto routing systems, artificial intelligence, computer games, among others. One of the most commonly used algorithms to solve path finding is Dijkstra’s algorithm and its variants [47], [55].

3) ROAD NETWORK

Defined as a network of roads that can be modeled as a graph, where the nodes represent a set of road intersections and the edges represents a set of road segments between two neighboring junctions. Being more specific, a taxi road network includes routes executed by different vehicles over different hours, where the time and the start and end point (nodes) of a taxi ride are saved. This information is later used to calculate optimal routes for taxi drivers [55]. Some related use cases are:

- Route query: It is defined as a path query usually used on maps, where the goal is obtaining a path to follow from a starting point to a destination point [100]. For example, in traffic networks a route query consists in to find a path between two locations. In social networks, a route query implies the search of the closest relationships (e.g. friendship) between two persons [107]. For example, in the context of a geographical network, where the nodes represent places in a city, and the edges represent the roads connecting the places, a common use-case is the search of paths allowing an efficient use of resources [108]. Several graph algorithms are used for route searching, including divide & conquer, breadth-first search, depth-first search, among others.
- Taxi trajectory: It means using a graph to represent streets, intersections, and locations in a city, and using paths to represent taxi trips [55].

4) SEMANTIC NETWORK

A semantic network is a graph structure used to represent knowledge by connecting nodes and arcs in specific patterns [109]. The nodes in a semantic network, referred to as units, represent an idea, event, situation, or any other object that typically has a complex structure. On the other hand, the directed links (edges) represent the relationships between the units [110]. In order to design semantic networks effectively, it is crucial to maintain a balance. This is because if the number of node types or link types increases, although the network becomes more expressive, it becomes increasingly challenging to implement inference engines on this type of graph [44].

TABLE 7. Classification of research papers based on their type.

Kind of paper	References
Use Case	[117], [33], [34], [119], [45], [36], [54], [120], [46], [58], [50], [35], [56], [57], [59], [47], [48], [44], [55], [51], [121], [122]
Practical	[37], [71], [123], [124], [97], [100], [125], [126], [127], [52], [76], [113], [128], [129], [130], [131], [98], [132], [92], [88], [133], [69], [115], [77], [134], [135], [73], [53], [72], [64], [136], [82], [137], [39], [83], [138], [85], [139], [121], [140]
Survey	[22]
Theoretical	[37], [63], [141], [96], [74], [124], [102], [142], [103], [91], [143], [144], [145], [84], [125], [75], [20], [95], [126], [146], [120], [127], [147], [76], [101], [129], [130], [148], [149], [132], [92], [89], [150], [151], [152], [68], [115], [104], [153], [154], [80], [97], [39], [138], [85], [155], [156], [140], [157], [158], [159]

TABLE 8. Classification by research papers based on their orientation.

Orientation	References
Define	[37], [63], [141], [33], [119], [102], [36], [103], [75], [126], [120], [147], [52], [76], [101], [129], [130], [148], [104], [80], [97], [39], [83], [138], [85], [155], [156], [158]
Implement	[117], [33], [96], [34], [45], [36], [97], [91], [100], [143], [144], [54], [20], [146], [46], [52], [76], [128], [130], [131], [149], [132], [92], [88], [150], [151], [50], [68], [115], [134], [53], [72], [64], [104], [136], [82], [154], [97], [39], [83], [48], [44], [138], [85], [22], [139], [122]
Optimize	[37], [117], [34], [123], [74], [97], [100], [84], [125], [54], [20], [95], [126], [127], [58], [52], [76], [113], [128], [131], [98], [149], [152], [77], [135], [73], [35], [56], [57], [59], [137], [47], [55], [121], [140], [157], [159]
Learn	[124], [145], [133]
Compare	[71], [142], [91], [101], [98], [89], [150], [69], [153], [51]

5) SOCIAL NETWORK

A social network can be defined as a finite set of actors and relationships that link the actors [52], [53]. A social network can be seen as a attributed graph where the users are the nodes, their profiles are attributes of nodes, and the relationships are represented by the edges. Numerous queries can be performed on a social network, such as searching for a connection between two individuals or analyzing interactions among people [22]. A related use case is:

- Social network analysis (SNA) Imagine a social network graph, where nodes symbolize actors and edges symbolize the relationships between these actors. Social network analysis focuses on solving intricate graph queries, such as determining the most influential actors and discovering the central actors within the network. These specific scenarios are addressed by utilizing graph measures like Page-Rank, Hubs, Centrality, Density, and others [111], [112], [113]. One important study related to SNA involves the examination of a network’s structure. The objective of this study is to develop

TABLE 9. Acronyms.

Acronym	Name
ACCT	Advanced Computing and Communication Technologies
ADC	Australasian Database Conference
AI Soc	AI & Society
ANN MATH ARTIF INTEL	Annals of Mathematics and Artificial Intelligence
ASONAM	Advances in Social Networks Analysis and Mining
BDCloud	Big Data and Cloud Computing
BNCOD	British National Conference on Databases
CCIS	Communications in Computer and Information Science
CEUR	CEUR Workshop Proceedings
CIE	Computability in Europe
CSCWD	Computer Supported Cooperative Work in Design
DASFAA	Database Systems for Advanced Applications
DBPL	Database Programming Languages
EDBT	Extending Database Technology
ER	Conceptual Modeling
ESWC	European Semantic Web Conference
FI	Fundamenta Informaticae
FSKD	Fuzzy Systems and Knowledge Discovery
GRADES	Graph Data Management Experiences and Systems
HOSC	Higher-Order and Symbolic Computation
I2CT	Conference on Convergence and its Application
ICDE	International Conference on Data Engineering
ICDM	Conference on Data Mining
ICDT	International Conference on Database Theory
ICFP	International Conference on Functional Programming
ICPC	International Conference on Program Comprehension
ICPS	ACM International Conference Proceeding Series
IEEE Access	IEEE Access
IISA	Information, Intelligence, Systems and Applications
IJCAI	International Joint Conference on Artificial Intelligence
Inf. Sci.	Information Sciences
Int. J. Appl. Earth Obs. Geoinf.	International Journal of Applied Earth Observation and Geoinformation
ISWC	International Semantic Web Conference
ITI	Information Technology Interfaces
J Comput Syst Sci	Journal of Computer and System Sciences
J. Comput.	Computer Journal
JACM	Journal of the ACM
JDM	Journal of Database Management
JHMSP	Journal of Information Hiding and Multimedia Signal Processing
LICS	Logic in Computer Science
LIPICs	Leibniz International Proceedings in Informatics
LMCS	Logical Methods in Computer Science
Math. Ann.	Mathematische Annalen
MJCS	Malaysian Journal of Computer Science
MPC	Mathematics of Program Construction
NATO	Combinatorial Programming: Methods and Applications
NDA	Network Data Analytics
NetSoft	Network Softwarization and Workshops
NSDI	Networked Systems Design and Implementation
Oper. Res. Lett.	Operations Research Letters
OS	Operating Systems
PODS	Principles of Database Systems
POPL	Principles of Programming Languages
Program. Comput. Softw.	Programming and Computer Software
PVLDB	Proceedings of the VLDB Endowment
RecSys	Recommender Systems
RR	Web Reasoning and Rule Systems
SIGMOD	Special Interest Group on Management of Data
SISY	Intelligent Systems and Informatics
SMAP	Semantic and Social Media Adaptation and Personalization
SSDBM	Scientific and Statistical Database Management
TCS	Theoretical Computer Science
TODS	Transactions on Database Systems
TOIS	Transactions on Information Systems
TOOLS	Technology of Object-Oriented

a better understanding of how the structure forms and which mechanisms influence the diffusion of information, contagion, community identification, and cohesive subgroup formation [114].

6) SOFTWARE LIFE CYCLE ACTIVITIES

A software life cycle can be represented by a directed graph where the nodes are activities and each edges represent an implication between activities. The advantage of representing a software life cycle as a graph is the possibility of using path expressions to represent specific cycles of software development [115].

7) ROUTING

This is a combinatorial problem in which we seek the best possible path from one point to another. The best path could be determined by various factors, such as length, time, and others. For example, given a road network containing cities and roads, a common routing query is to find the optimal path between the cities [116]. Other use case for routing, is the search of an optimal path for information routing on a communication network. Routing problems are usually solved by using shortest path algorithms [117]. Some related use cases are:

- **Route searching:** In general, this concept is related to the problems where is needed to search all the routes between two or more nodes in a graph. Route searching is a use-case in geographical information systems navigation, computer games, virtual reality, among others [47].
- **Traffic Routing:** It pertains to a set of routing problems in which the objective is to discover the most efficient route between two points [116]. For example, given a graph where the nodes represent geographical locations, paths are useful to study congestion and security problems [56]. Another example is a graph of broadcast information between computers, where the computers or network devices are the nodes, and the connections between them are the edges. In this context, a use-case concerns the distribution of the traffic throughout the network by using the best possible path [117].

VI. CONCLUSION

This paper was carried out to identify and establish the current state of knowledge on path querying in graph databases. To conduct a study using a standardized framework, we performed a systematic mapping study (SMS) in which we reviewed four digital libraries and collected research papers published from 1970 to 2022.

After the gathering phase, we obtained a total of 97 relevant papers among the 17 selected research keywords. The results provide both quantitative and qualitative information on path querying in graph databases. Our quantitative analysis shows that most of the research focuses on path query, especially from 2000 to 2022. However, there is little research on

specific areas like path manipulation, where we found no related work. Another interesting finding was related to the year when the first related paper was established, which was around the 70's. It gained popularity at the end of the 90's, where the three most popular terms were "path querying", "path discovery" and "path pattern". The last term has gained interest because it poses a challenge for future graph query languages.

On the other hand, our qualitative analysis reveals that most of the research is about implementing methods for path querying, with optimization techniques as the second most common topic. We also found papers in significant conferences such as ICDE, PODS and SIGMOD. Another interesting point is related to query languages that use paths. In particular, we identified some graph query languages like G-CORE [39] and Cypher [40], Semantic Web oriented query languages like SPARQL [41], and Web oriented query languages like GraphQL [42].

For future research in the topic of graph querying, we suggest that the investigation could focus on areas such as path manipulation, path indexing, path query optimization, machine learning for path querying, and others. This is based on the ongoing work on the standardization of the graph query languages, that is now a global standards project carried out by the GQL standards committee [118].

Finally we trust that this article can serve as a central source for bibliographic citations, providing quick and easy access to research papers related to path querying. Moreover, it can encourage further research on path manipulation, which includes the development of concepts such as path algebra, path operations, path insertion, path deletion, path updating, path construction, and more.

APPENDIX A PAPER CLASSIFICATION

See Tables 7 and 8.

APPENDIX B ACRONYMS

See Table 9.

REFERENCES

- [1] R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Comput. Surveys*, vol. 40, no. 1, pp. 1–39, Feb. 2008.
- [2] R. Angles, J. Reutter, and H. Voigt, "Graph query languages," in *Encyclopedia Big Data Technologies*, S. Sakr and A. Zomaya, Eds. Cham, Switzerland: Springer, 2018, pp. 1–8.
- [3] S. Sakr et al., "The future is big graphs: A community view on graph processing systems," *Commun. ACM*, vol. 64, no. 9, pp. 62–71, Aug. 2021.
- [4] M. V. Vieira, B. M. Fonseca, R. Damazio, P. B. Golgher, D. D. C. Reis, and B. Ribeiro-Neto, "Efficient search ranking in social networks," in *Proc. 16th ACM Conf. Conf. Inf. Knowl. Manage.*, Nov. 2007, pp. 563–572.
- [5] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: Membership, growth, and evolution," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: Association for Computing Machinery, 2006, pp. 44–54.

- [6] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, vol. 11. New York, NY, USA: ACM, Aug. 2003, pp. 137–146.
- [7] M. Gong, G. Li, Z. Wang, L. Ma, and D. Tian, "An efficient shortest path approach for social networks based on community structure," *CAAI Trans. Intell. Technol.*, vol. 1, no. 1, pp. 114–123, Jan. 2016.
- [8] S. Srivastava and A. K. Singh, "Fraud detection in the distributed graph database," *Cluster Comput.*, vol. 26, no. 1, pp. 515–537, Feb. 2023.
- [9] Z. Chen, S. Zhang, X. Zeng, M. Mei, X. Luo, and L. Zheng, "Parallel path detection for fraudulent accounts in banks based on graph analysis," *PeerJ Comput. Sci.*, vol. 9, p. e1749, Dec. 2023.
- [10] B. P. Kelley, B. Yuan, F. Lewitter, R. Sharan, B. R. Stockwell, and T. Ideker, "PathBLAST: A tool for alignment of protein interaction networks," *Nucleic Acids Res.*, vol. 32, pp. W83–W88, Jul. 2004.
- [11] T. Shlomi, D. Segal, E. Ruppin, and R. Sharan, "QPath: A method for querying pathways in a protein-protein interaction network," *BMC Bioinf.*, vol. 7, no. 1, p. 199, Dec. 2006.
- [12] Q. Yang and S.-H. Sze, "Path matching and graph matching in biological networks," *J. Comput. Biol.*, vol. 14, no. 1, pp. 56–67, Jan. 2007.
- [13] A. Deutsch, N. Francis, A. Green, K. Hare, B. Li, L. Libkin, T. Lindaaker, V. Marsault, W. Martens, J. Michels, F. Murlak, S. Plantikow, P. Selmer, O. van Rest, H. Voigt, D. Vrgoč, M. Wu, and F. Zemke, "Graph pattern matching in GQL and SQL/PGQ," in *Proc. Int. Conf. Manage. Data*, vol. 46, no. 1. New York, NY, USA: ACM, 2022, pp. 2246–2258.
- [14] N. Francis, A. Gheerbrant, P. Guagliardo, L. Libkin, V. Marsault, W. Martens, F. Murlak, L. Peterfreund, A. Rogova, and D. Vrgoč, "A researcher's digest of GQL," in *Proc. Leibniz Int. Inform. (LIPIcs)*, Dagstuhl, Germany, 2023, pp. 1:1–1:22, Art. no. 1.
- [15] R. Angles, "A comparison of current graph database models," in *Proc. IEEE 28th Int. Conf. Data Eng. Workshops*, Apr. 2012, pp. 171–177.
- [16] S.-T. Wang, J. Jin, P. Rivett, and A. Kitazawa, "Technical survey graph databases and applications," *Int. J. Semantic Comput.*, vol. 9, no. 4, pp. 523–545, Dec. 2015.
- [17] K. Rabuzin and M. Šestak, "Graph database management systems: The past, the present, and the future," in *Encyclopedia of Information Science and Technology*, 5th ed., M. Khosrow-Pour, Ed. Hershey, PA, USA: IGI Global, 2021, pp. 778–790.
- [18] S. Timón-Reina, M. Rincón, and R. Martínez-Tomás, "An overview of graph databases and their applications in the biomedical domain," *Database*, vol. 2021, pp. 1–22, May 2021.
- [19] P. Barceló, L. Libkin, and J. L. Reutter, "Querying graph patterns," in *Proc. Symp. Princ. Database Syst. (PODS)*. New York, NY, USA: ACM, 2011, pp. 199–210.
- [20] G. Bagan, A. Bonifati, and B. Groz, "A trichotomy for regular simple path queries on graphs," in *Proc. 32nd ACM SIGMOD-SIGACT-SIGAI Symp. Princ. Database Syst.* New York, NY, USA: ACM, Jun. 2013, pp. 261–272.
- [21] A. Bonifati and S. Dumbava, "Graph queries: From theory to practice," *ACM SIGMOD Rec.*, vol. 47, no. 4, pp. 5–16, May 2019.
- [22] Y. Wang, Y. Li, J. Fan, C. Ye, and M. Chai, "A survey of typical attributed graph queries," in *World Wide Web*, vol. 24. Berlin, Germany: Springer, 2021, pp. 297–346.
- [23] P. T. Wood, "Query languages for graph databases," *ACM SIGMOD Rec.*, vol. 41, no. 1, pp. 50–60, Apr. 2012.
- [24] R. Angles, M. Arenas, P. Barceló, A. Hogan, J. Reutter, and D. Vrgoč, "Foundations of modern query languages for graph databases," *ACM Comput. Surveys*, vol. 50, no. 5, pp. 1–40, Sep. 2017.
- [25] C. Sharma, R. Sinha, and K. Johnson, "Practical and comprehensive formalisms for modelling contemporary graph query languages," *Inf. Syst.*, vol. 102, Dec. 2021, Art. no. 101816.
- [26] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," EBS, Univ. Durham, Durham, U.K., Tech. Rep. 1051, 2007.
- [27] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proc. Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, vol. 17. Swindon, U.K.: BCS Learning & Development Ltd., 2008, pp. 68–77.
- [28] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015.
- [29] (2021). *ACM Digital Library*. [Online]. Available: <https://dl.acm.org>
- [30] (2021). *Advancing Technology for Humanity*. [Online]. Available: <https://ieeexplore.ieee.org/Xplore/home.jsp>
- [31] (2021). *Springer Link*. [Online]. Available: <https://link.springer.com/>
- [32] (2021). *Scopus*. [Online]. Available: <https://www.scopus.com>
- [33] X. Huang, Z. Bao, S. B. Davidson, T. Milo, and X. Yuan, "Answering regular path queries on workflow provenance," in *Proc. IEEE 31st Int. Conf. Data Eng.*, Apr. 2015, pp. 375–386.
- [34] Y.-C. Chen and C. Lee, "Skyline path queries with aggregate attributes," *IEEE Access*, vol. 4, pp. 4690–4706, 2016.
- [35] J. Pan, C. Wang, and T. Sung, "Road network modeling with layered abstraction for path discovery in vehicle navigation systems," *J. Inf. Hiding Multim. Signal Process.*, vol. 7, no. 4, pp. 791–801, 2016.
- [36] T. Johnson, Y. Kanza, L. V. S. Lakshmanan, and V. Shkapenyuk, "Nepal: A path query language for communication networks," in *Proc. Workshop Netw. Data Analytics (NDA)*. New York, NY, USA: ACM, 2016, pp. 1–8.
- [37] Y. Bai, C. Wang, X. Ying, M. Wang, and Y. Gong, "Path pattern query processing on large graphs," in *Proc. IEEE 4th Int. Conf. Big Data Cloud Comput.*, Dec. 2014, pp. 767–774.
- [38] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi, "Reasoning on regular path queries," *ACM SIGMOD Rec.*, vol. 32, no. 4, pp. 83–92, Dec. 2003.
- [39] R. Angles, M. Arenas, P. Barcelo, P. Boncz, G. Fletcher, C. Gutierrez, T. Lindaaker, M. Paradies, S. Plantikow, J. Sequeda, O. van Rest, and H. Voigt, "G-core: A core for future graph query languages," in *Proc. Int. Conf. Manage. Data (SIGMOD)*. New York, NY, USA: ACM, 2018, pp. 1421–1432.
- [40] Neo4j. (2020). *Neo4j Graph Platform The Leader in Graph Databases*. [Online]. Available: <https://neo4j.com/>
- [41] E. Prud'hommeaux and A. Seaborne. (Jan. 15, 2008). *SPARQL Query Language for RDF (W3C Recommendation)*. [Online]. Available: <https://www.w3.org/TR/rdf-sparql-query/>
- [42] I Facebook. (2016). *GraphQL | A Query Language for Your API*. [Online]. Available: <https://graphql.org/>
- [43] A. Bonifati, G. Fletcher, H. Voigt, and N. Yakovets, *Querying Graphs (Synthesis Lectures on Data Management)*. San Rafael, CA, USA: Morgan & Claypool, 2018.
- [44] R. Ayres, "Databases for criminal intelligence analysis: Knowledge representation issues," *AI Society*, vol. 11, nos. 1–2, pp. 18–35, 1997.
- [45] G.-Q. Wu, X. Wu, X.-G. Hu, H.-G. Li, Y. Liu, and R.-G. Xu, "Web news extraction based on path pattern mining," in *Proc. 6th Int. Conf. Fuzzy Syst. Knowl. Discovery*, vol. 7, Aug. 2009, pp. 612–617.
- [46] S. M. Dintelman and A. T. Maness, "An implementation of a query language supporting path expressions," in *Proc. ACM SIGMOD Int. Conf. Manage. Data SIGMOD*. New York, NY, USA: ACM, 1982, pp. 87–93.
- [47] B. Sobota, C. Szabo, and J. Perhac, "Using path-finding algorithms of graph theory for route-searching in geographical information systems," in *Proc. Int. Symp. Intell. Syst. Inform. (SISY)*, 2008, pp. 1–6.
- [48] N. C. Abay, A. Mutlu, and P. Karagoz, "A path-finding based method for concept discovery in graphs," in *Proc. 6th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2015, pp. 1–6.
- [49] S. R. Kimpleton and G. M. Schneider, "Computer communication networks: Approaches, objectives, and performance considerations," *ACM Comput. Surveys*, vol. 7, no. 3, pp. 129–173, Sep. 1975.
- [50] S. Narayana, M. T. Arashloo, J. Rexford, and D. Walker, "Compiling path queries," in *Proc. USENIX Conf. Networked Syst. Design Implement. (NSDI)*. Berkeley, CA, USA: USENIX Association, 2016, pp. 207–222.
- [51] W. Nawaz, K.-U. Khan, and Y.-K. Lee, "CORE analysis for efficient shortest path traversal queries in social graphs," in *Proc. IEEE 4th Int. Conf. Big Data Cloud Comput.*, Dec. 2014, pp. 363–370.
- [52] Y. Bai, C. Wang, Y. Ning, H. Wu, and H. Wang, "G-path: Flexible path pattern query on large graphs," in *Proc. 22nd Int. Conf. World Wide Web*. New York, NY, USA: ACM, May 2013, pp. 333–336.
- [53] P. Brodka, P. Stawiak, and P. Kazienko, "Shortest path discovery in the multi-layered social network," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining*, Jul. 2011, pp. 497–501.
- [54] M. Hua and J. Pei, "Probabilistic path queries in road networks: Traffic uncertainty aware path selection," in *Proc. Int. Conf. Extending Database Technol. (EDBT)*. New York, NY, USA: ACM, Mar. 2010, pp. 347–358.
- [55] Q. Li, Z. Zeng, T. Zhang, J. Li, and Z. Wu, "Path-finding through flexible hierarchical road networks: An experiential approach using taxi trajectory data," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 13, no. 1, pp. 110–119, Feb. 2011.

- [56] A. Dahiya, Madhu, and N. Bansal, "Path discovery in vehicular ad hoc networks," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Technol.*, Jan. 2012, pp. 551–555.
- [57] V. Bhanumathi and R. Dhanasekaran, "Path discovery and selection for energy efficient routing with transmit power control in MANET," *Malaysian J. Comput. Sci.*, vol. 26, no. 2, pp. 124–139, 2013.
- [58] C. Costa, X. Ge, and P. K. Chrysanthis, "CAPRIO: Graph-based integration of indoor and outdoor data for path discovery," *Proc. VLDB Endowment*, vol. 12, no. 12, pp. 1878–1881, Aug. 2019.
- [59] R. H. Campbell and A. N. Habermann, "The specification of process synchronization by path expressions," in *Operating Systems*. Berlin, Germany: Springer, 1974, pp. 89–102.
- [60] G. Rote, "Path problems in graphs," in *Computational Graph Theory*, G. Tinhofer, E. Mayr, H. Noltemeier, and M. M. Syslo, Eds. Vienna, Austria: Springer, 1990, pp. 155–189.
- [61] A. Schrijver, "On the history of the shortest path problem," *Documenta Math.*, vol. 17, no. 1, pp. 155–167, 2012.
- [62] A. Efentakis, D. Pfoser, and A. Voisard, "Efficient data management in support of shortest-path computation," in *Proc. 4th ACM SIGSPATIAL Int. Workshop Comput. Transp. Sci.* New York, NY, USA: ACM, Nov. 2011, pp. 28–33.
- [63] R. Manger, "A new path algebra for finding paths in graphs," in *Proc. Int. Conf. Technol. Interfaces (ITI)*, vol. 1, 2004, pp. 657–662.
- [64] J. Hong, Y. Han, and Y.-K. Lee, "A RDBMS based efficient method for shortest path searching over large graphs considering constraint conditions," in *Proc. IT Converg. Appl.* South Korea: Science & Engineering Research Support soCieTy, Nov. 2013, pp. 96–100.
- [65] Y. Yuan, X. Lian, G. Wang, Y. Ma, and Y. Wang, "Constrained shortest path query in a large time-dependent graph," *Proc. VLDB Endowment*, vol. 12, no. 10, pp. 1058–1070, Jun. 2019.
- [66] G. F. Italiano, "Amortized efficiency of a path retrieval data structure," *Theor. Comput. Sci.*, vol. 48, pp. 273–281, 1986.
- [67] N. Jing, Y.-W. Huang, and E. A. Rundensteiner, "Hierarchical optimization of optimal path finding for transportation applications," in *Proc. 5th Int. Conf. Inf. Knowl. Manage. CIKM*. New York, NY, USA: ACM, 1996, pp. 261–268.
- [68] J. Sosnowska and O. Skibski, "Path evaluation and centralities in weighted graphs—An axiomatic approach," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2018, pp. 3856–3862.
- [69] D. Taniar, "Forward vs. reverse traversal in path expression query processing," in *Proc. Technol. Object-Oriented Lang. (TOOLS)*, 1998, pp. 127–140.
- [70] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall Series in Artificial Intelligence), 3rd ed. London, U.K.: Pearson Education, 2010.
- [71] P. Bouros and Y. Vassiliou, "Evaluating path queries over route collections," in *Proc. IEEE 26th Int. Conf. Data Eng. Workshops (ICDEW)*. Los Alamitos, CA, USA: IEEE Computer Society, Mar. 2010, pp. 325–328.
- [72] C. Szepesvári, "Shortest path discovery problems: A framework, algorithms and experimental results," in *Proc. Conf. Artif. Intell.* San Jose, CA, USA: American Association for Artificial Intelligence, Jan. 2004, pp. 550–555.
- [73] J. Hong, K. Park, Y. Han, M. K. Rassel, D. Vonvou, and Y.-K. Lee, "Disk-based shortest path discovery using distance index over large dynamic graphs," *Inf. Sci.*, vols. 382–383, pp. 201–215, Mar. 2017.
- [74] M. Fernandez and D. Suciu, "Optimizing regular path expressions using graph schemas," in *Proc. 14th Int. Conf. Data Eng.*, Feb. 1998, pp. 14–23.
- [75] P. Barceló, L. Libkin, A. W. Lin, and P. T. Wood, "Expressive languages for path queries over graph-structured data," *ACM Trans. Database Syst.*, vol. 37, no. 4, pp. 1–46, Dec. 2012.
- [76] Y. Bai, C. Wang, and X. Ying, "Para-G: Path pattern query processing on large graphs," *World Wide Web*, vol. 20, no. 3, pp. 515–541, May 2017.
- [77] R. E. Tarjan, "Fast algorithms for solving path problems," *J. ACM*, vol. 28, no. 3, pp. 594–614, Jul. 1981.
- [78] S. Harris and A. Seaborne. (Mar. 21, 2013). *SPARQL 1.1 Query Language (W3C Recommendation)*. [Online]. Available: <https://www.w3.org/TR/sparql11-query/>
- [79] A. Seaborne. (2010). *SPARQL 1.1 Property Paths*. [Online]. Available: <http://www.w3.org/TR/sparql11-property-paths/>
- [80] X. Zhang, Z. Feng, X. Wang, G. Rao, and W. Wu, "Context-free path queries on RDF graphs," in *Proc. Semantic Web (ISWC)*. Cham, Switzerland: Springer, 2016, pp. 632–648.
- [81] K. Losemann and W. Martens, "The complexity of regular expressions and property paths in SPARQL," *ACM Trans. Database Syst.*, vol. 38, no. 4, pp. 1–39, Nov. 2013.
- [82] R. Hamill and N. Martin, "Database support for path query functions," in *Key Technologies for Data Management*, vol. 3112. Berlin, Germany: Springer, 2004, pp. 84–99.
- [83] O. Van Rest, S. Hong, J. Kim, X. Meng, and H. Chafi, "PGQL: A property graph query language," in *Proc. Int. Workshop Graph Data Manage. Experiences Syst. (GRADES)*. New York, NY, USA: ACM, 2016, pp. 1–6.
- [84] S. Abiteboul and V. Vianu, "Regular path queries with constraints," in *Proc. Symp. Princ. Database Syst. (PODS)*. New York, NY, USA: ACM, 1997, pp. 122–133.
- [85] J. Hellings, "Explaining results of path queries on graphs: Single-path results for context-free path queries," in *Communications in Computer and Information Science*, vol. 1281. Berlin, Germany: Springer, 2020, pp. 84–98.
- [86] J. W. Rahayu, D. Taniar, and X. Lu, "Aggregation query model for OODBMS," in *Proc. Int. Conf. Tools Pacific, Objects Internet, Mobile Embedded Appl. (CRPIT)*. Sydney, NSW, Australia: Australian Computer Society, 2002, pp. 143–150.
- [87] S. Flesca, F. Furfaro, and S. Greco, "Weighted path queries on semistructured databases," *Inf. Comput.*, vol. 204, no. 5, pp. 679–696, May 2006.
- [88] I. Papadakis, S. Kazanas, and M. Stefanidakis, "Navigational queries as property path queries employing the kleene star operator," in *Proc. 11th Int. Workshop Semantic Social Media Adaptation Personalization (SMAP)*, Oct. 2016, pp. 7–12.
- [89] P. Barcelo, G. Fontaine, and A. W. Lin, "Expressive path queries on graph with data," *Log. Methods Comput. Sci.*, vol. 11, no. 4, pp. 1–39, Oct. 2015.
- [90] J. L. Reutter and D. Vrgoč, "Navigational and rule-based languages for graph databases," in *Proc. Reasoning Web, Log. Found. Knowl. Graph Construct. Query Answering 12th Int. Summer School*, Aberdeen, U.K., Cham, Switzerland: Springer, 2017, pp. 90–123.
- [91] P. B. Baeza, "Querying graph databases," in *Proc. Symp. Princ. Database Syst. (PODS)*. New York, NY, USA: ACM, 2013, pp. 175–188.
- [92] H. Wu, Y. Huang, J. Cheng, J. Li, and Y. Ke, "Reachability and time-based path queries in temporal graphs," in *Proc. Int. Conf. Data Eng. (ICDE)*, 2016, pp. 145–156.
- [93] Z. Zhang, J. X. Yu, L. Qin, Q. Zhu, and X. Zhou, "I/O cost minimization: Reachability queries processing over massive graphs," in *Proc. Int. Conf. Extending Database Technol. (EDBT)*. New York, NY, USA: ACM, 2012, pp. 468–479.
- [94] J. X. Yu and J. Cheng, "Graph reachability queries: A survey," in *Managing and Mining Graph Data*. Boston, MA, USA: Springer, 2010, pp. 181–215.
- [95] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi, "Rewriting of regular expressions and regular path queries," *J. Comput. Syst. Sci.*, vol. 64, no. 3, pp. 443–465, May 2002.
- [96] D. Figueira and L. Libkin, "Path logics for querying graphs: Combining expressiveness and efficiency," in *Proc. 30th Annu. ACM/IEEE Symp. Log. Comput. Sci.* Washington, DC, USA: IEEE Computer Society, Jul. 2015, pp. 329–340.
- [97] F. Tetzl, "Don't walk twice: Path sharing for regular path queries on large graphs," in *Proc. ACM Int. Conf. Manage. Data*. New York, NY, USA: ACM, May 2017, pp. 25–27.
- [98] J. Kuijpers, G. Fletcher, N. Yakovets, and T. Lindaaker, "An experimental study of context-free path query evaluation methods," in *Proc. Int. Conf. Stat. Database Manag. (SSDBM)*. New York, NY, USA: ACM, 2019, pp. 121–132.
- [99] M. L. Schmid, "Conjunctive regular path queries with string variables," in *Proc. 39th ACM SIGMOD-SIGACT-SIGAI Symp. Princ. Database Syst.* New York, NY, USA: ACM, Jun. 2020, pp. 361–374.
- [100] E. P. F. Chan and H. Lim, "Optimization and evaluation of shortest path queries," *VLDB J.*, vol. 16, no. 3, pp. 343–369, May 2007.
- [101] F. Bry, T. Furche, and B. Linse, "The perfect match: RPL and RDF rule languages," in *Proc. Int. Conf. Web Reasoning Rule Syst. (RR)*. Berlin, Germany: Springer-Verlag, 2009, pp. 227–241.
- [102] D. Colazzo and C. Sartiani, "Typing regular path query languages for data graphs," in *Proc. Symp. Database Program. Lang. (DBLP)*. New York, NY, USA: ACM, 2015, pp. 69–78.
- [103] N. Fuhr and K. Großjohann, "XIRQL: An XML query language based on information retrieval concepts," *ACM Trans. Inf. Syst.*, vol. 22, no. 2, pp. 313–356, Apr. 2004.

- [104] M. Gondran, "Path algebra and algorithms," in *Combinatorial Programming: Methods and Applications*. Dordrecht, The Netherlands: Springer, 1975, pp. 137–148.
- [105] P. Weinberg, J. Groff, and A. Oppel, *SQL, The Complete Reference*. New York, NY, USA: McGraw-Hill, 2010.
- [106] J. Zhu, Q. Meng, W. Chen, and Z. Ma, "Interpreting the basis path set in neural networks," *J. Syst. Sci. Complex.*, vol. 34, no. 6, pp. 2155–2167, Dec. 2021.
- [107] Y. Yang, H. Zhang, H. Gao, Q. Hu, and X. Wang, "An efficient index method for the optimal path query over multi-cost networks," *World Wide Web*, vol. 24, pp. 697–719, Mar. 2020.
- [108] R. Pries, D. Staehle, B. Staehle, and P. Tran-Gia, "On optimization of wireless mesh networks using genetic algorithms," *Int. J. Advance Internet Technol.*, vols. 1–2, pp. 13–28, Jan. 2010.
- [109] J. F. Sowa, "Semantic networks," in *Encyclopedia of Cognitive Science*. Hoboken, NJ, USA: Wiley, 2006, pp. 1–15.
- [110] F. Lehmann, "Semantic networks," *Comput. Math. Appl.*, vol. 23, no. 2, pp. 1–50, 1992.
- [111] M. Oliveira and J. Gama, "An overview of social network analysis," *WIREs Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 99–115, 2012.
- [112] A. Alamsyah, B. Rahardjo, and Kuspriyanto, "Social network analysis taxonomy based on graph representation," 2021, *arXiv:2102.08888*.
- [113] M. Przyjacieli-Zablocki, A. Schätzle, T. Hornung, and G. Lausen, "RDF-Path: Path query processing on large RDF graphs with MapReduce," in *Proc. Semantic Web, ESWC Workshops*, vol. 7117. Berlin, Germany: Springer, 2012, pp. 50–64.
- [114] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications* (Structural Analysis in the Social Sciences). Cambridge, U.K.: Cambridge Univ. Press, 1994.
- [115] A. Bertolino and M. Marre, "Deriving path expressions recursively," in *Proc. Workshop Program Comprehension (WPC)*, 1993, pp. 177–185.
- [116] R. Bellman, "On a routing problem," *Quart. Appl. Math.*, vol. 16, no. 1, pp. 87–90, 1958.
- [117] S. Gao, S. Shrivastava, K. Ogan, Y. Xin, and I. Baldin, "Evaluating path query mechanisms as a foundation for SDN network control," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 28–36.
- [118] I JCC Consulting. (2020). *Graph Query Language GQL*. [Online]. Available: <https://www.gqlstandards.org/home>
- [119] J. Kim, T. Han, and S. K. Lee, "Visualization of path expressions in a visual object-oriented database query language," in *Proc. 6th Int. Conf. Adv. Syst. Adv. Appl.*, 1999, pp. 99–108.
- [120] P. E. Lauer and R. H. Campbell, "A description of path expressions by Petri nets," in *Proc. Symp. Princ. Program. Lang. (POPL)*. New York, NY, USA: ACM, 1975, pp. 95–105.
- [121] X. Li, B. Wang, T. Qiu, G. Yu, and N. Wang, "Constructing effective caches of shortest path queries on road networks," *IEEE Access*, vol. 8, pp. 193777–193789, 2020.
- [122] X. Lin, Y. Wang, Y. Tian, and L. Wu, "Path querying system on mobile devices," in *Remote Sensing and Space Technology for Multidisciplinary Research and Applications*, vol. 6199, Q. Tong, X. Chen, A. Huang, and W. Gao, Eds. Bellingham, WA, USA: SPIE, 2006, Art. no. 61990N.
- [123] S. Tao, L. Zhang, and H. Ma, "An automatic generation method for path expressions of CPN model focus on tested behaviors," in *Proc. Int. Conf. Comput. Supported Cooperat. Work Design ((CSCWD))*, 2018, pp. 31–38.
- [124] P. Ding, G. Liu, P. Zhao, A. Liu, Z. Li, and K. Zheng, "Reinforcement learning based Monte Carlo tree search for temporal path discovery," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 140–149.
- [125] M. Nolè and C. Sartiani, "Regular path queries on massive graphs," in *Proc. 28th Int. Conf. Scientific Stat. Database Manage.* New York, NY, USA: ACM, Jul. 2016, pp. 1–12.
- [126] A. Matono, T. Amagasa, M. Yoshikawa, and S. Uemura, "A path-based relational RDF database," in *Proc. Australas. Database Conf. (ADC)*, vol. 39. Sydney, NSW, Australia: Australian Computer Society, 2005, pp. 95–103.
- [127] V. Christophides, S. Cluet, and G. Moerkotte, "Evaluating queries with generalized path expressions," *ACM SIGMOD Rec.*, vol. 25, no. 2, pp. 413–422, Jun. 1996.
- [128] V. Pan and J. Reif, "Parallel nested dissection for path algebra computations," *Operations Res. Lett.*, vol. 5, no. 4, pp. 177–184, Oct. 1986.
- [129] V. Naudžiunas and T. G. Griffin, "A domain-specific language for the specification of path algebras," in *Proc. Workshop Automated Theory Eng. (ATE)*, vol. 760, Jul. 2011, pp. 46–57.
- [130] Y. Wu, D. Van Gucht, M. Gyssens, and J. Paredaens, "A study of a positive fragment of path queries: Expressiveness, normal form and minimization," *Comput. J.*, vol. 54, no. 7, pp. 1091–1118, Jul. 2011.
- [131] M. Nolè and C. Sartiani, "Processing regular path queries on Giraph," in *Proc. Workshop EDBT/ICDT Joint Conf.*, vol. 1133, Mar. 2014, pp. 37–40.
- [132] G. Grahne and A. Thomo, "Regular path queries under approximate semantics," *Ann. Math. Artif. Intell.*, vol. 46, nos. 1–2, pp. 165–190, Feb. 2006.
- [133] A. Bonifati, R. Ciucanu, and A. Lemay, "Learning path queries on graph databases," in *Proc. Int. Conf. Extending Database Technol. (EDBT)*, Mar. 2015, pp. 109–120.
- [134] M. Junkkari, J. Vainio, K. Iltanen, P. Arvola, H. Kari, and J. Kekäläinen, "Path expressions in SQL: A user study on query formulation," *J. Database Manage.*, vol. 27, no. 3, pp. 1–22, Jul. 2016.
- [135] A. Zhao, G. Liu, B. Zheng, Y. Zhao, and K. Zheng, "Temporal paths discovery with multiple constraints in attributed dynamic graphs," *World Wide Web*, vol. 23, no. 1, pp. 313–336, Jan. 2020.
- [136] J. Shi, G. Liu, A. Zhao, A. Liu, Z. Li, and K. Zheng, "A multi-constrained temporal path query system," in *Advances in Conceptual Modeling*. Cham, Switzerland: Springer, 2018, pp. 39–43.
- [137] J. Hong, H. Kim, W. Nawaz, K. Park, B. S. Jeong, and Y. K. Lee, "Distributed K-distance indexing approach for efficient shortest path discovery on large graphs," in *Database Systems for Advanced Applications*. Berlin, Germany: Springer, 2014, pp. 75–88.
- [138] M. Kyriakidi, G. Koutrika, and Y. Ioannidis, "Recommendations as graph explorations," in *Proc. 14th ACM Conf. Recommender Syst.* New York, NY, USA: ACM, Sep. 2020, pp. 289–298.
- [139] C. Aebeloe, G. Montoya, V. Setty, and K. Hose, "Discovering diversified paths in knowledge bases," *Proc. VLDB Endowment*, vol. 11, no. 12, pp. 2002–2005, Aug. 2018.
- [140] R. Azimov and S. Grigorev, "Context-free path querying by matrix multiplication," in *Proc. Int. Workshop Graph Data Manage. Experiences Syst. Netw. Data Analytics (GRADES-NDA)*. New York, NY, USA: ACM, 2018, pp. 1–10.
- [141] M. A. Rodriguez and P. Neubauer, "A path algebra for multi-relational graphs," in *Proc. IEEE 27th Int. Conf. Data Eng. Workshops*, Apr. 2011, pp. 128–131.
- [142] J. Hellings, B. Kuijpers, J. Van den Bussche, and X. Zhang, "Walk logic as a framework for path query languages on graph databases," in *Proc. 16th Int. Conf. Database Theory*. New York, NY, USA: ACM, Mar. 2013, pp. 117–128.
- [143] L. Libkin and D. Vrgoč, "Regular path queries on graphs with data," in *Proc. Int. Conf. Database Theory (ICDT)*. New York, NY, USA: ACM, 2012, pp. 74–85.
- [144] W. Martens and T. Trautner, "Dichotomies for evaluating simple regular path queries," *ACM Trans. Database Syst.*, vol. 44, no. 4, pp. 1–46, Oct. 2019.
- [145] S. Staworko and P. Wiecek, "Learning twig and path queries," in *Proc. 15th Int. Conf. Database Theory*. New York, NY, USA: ACM, Mar. 2012, pp. 140–154.
- [146] K. Losemann and W. Martens, "The complexity of evaluating path expressions in SPARQL," in *Proc. 31st ACM SIGMOD-SIGACT-SIGAI Symp. Princ. Database Syst.* New York, NY, USA: ACM, May 2012, pp. 101–112.
- [147] M. Murata, "Extended path expressions of XML," in *Proc. 20th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.* New York, NY, USA: ACM, May 2001, pp. 126–137.
- [148] G. Grahne, A. Thomo, and W. W. Wadge, "Preferential regular path queries," *Fundam. Inf.*, vol. 89, nos. 2–3, pp. 259–288, Apr. 2008.
- [149] M. Bienvenu, M. Ortiz, and M. Šimkus, "Conjunctive regular path queries in lightweight description logics," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*. Washington, DC, USA: AAAI Press, 2013, pp. 761–767.
- [150] W. Martens and T. Trautner, "Evaluation and enumeration problems for regular path queries," in *Proc. Int. Conf. Database Theory (ICDT)*, B. Kimelfeld and Y. Amsterdamer, Eds. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, vol. 98, 2018, pp. 19:1–19:21.

- [151] T. Gagie, M. He, and G. Navarro, "Path queries on functions," *Theor. Comput. Sci.*, vol. 770, pp. 34–50, May 2019.
- [152] O. De Moor, D. Lacey, and E. Van Wyk, "Universal regular path queries," *Higher-Order Symbolic Comput.*, vol. 16, nos. 1–2, pp. 15–35, 2003.
- [153] L. Beaudou, F. Foucaud, F. Madelaine, L. Nourine, and G. Richard, "Complexity of conjunctive regular path query homomorphisms," in *Computing With Foresight and Industry*, vol. 11558. Cham, Switzerland: Springer, 2019, pp. 108–119.
- [154] Y. A. Liu and F. Yu, "Solving regular path queries," in *Mathematics of Program Construction*, vol. 2386. Berlin, Germany: Springer, 2002, pp. 195–208.
- [155] J. Hellings, M. Gyssens, Y. Wu, D. Van Gucht, J. Van den Bussche, S. Vansummeren, and G. H. L. Fletcher, "Relative expressive power of downward fragments of navigational query languages on trees and chains," in *Proc. Symp. Database Program. Lang. (DBPL)*. New York, NY, USA: ACM, 2015, pp. 59–68.
- [156] E. L. Wilson, R. L. Hemminger, and M. D. Plummer, "A family of path properties for graphs," *Mathematische Annalen*, vol. 197, no. 2, pp. 107–122, Jun. 1972.
- [157] A. Morihata, K. Matsuzaki, and M. Takeichi, "Write it recursively: A generic framework for optimal path queries," in *Proc. Int. Conf. Funct. Program. (ICFP)*. New York, NY, USA: ACM, 2008, pp. 169–178.
- [158] R. Azimov and S. Grigorev, "Path querying with conjunctive grammars by matrix multiplication," *Program. Comput. Softw.*, vol. 45, no. 7, pp. 357–364, Dec. 2019.
- [159] Z. Zhang, Z. Wang, Z. Ding, H. Wang, K. Liu, and W. Wang, "A single source optimal reachable path query method for gene regulatory network," in *Proc. 2nd Int. Conf. Comput. Data Sci.* New York, NY, USA: ACM, Jan. 2021, pp. 1–5.



ROBERTO GARCÍA received the bachelor's degree in computer science from Universidad de Talca, Chile, in 2014, where he is currently pursuing the Ph.D. degree with the Program in Engineering Systems, Faculty of Engineering. Since March 2019, he belongs to the Engineering Systems Doctoral Program, Universidad de Talca. His research interests include graph query languages and benchmarking of graph databases.



RENZO ANGLES received the bachelor's degree in systems engineering from Universidad Católica de Santa Mara, Arequipa, Peru, and the Ph.D. degree in computer science from Universidad de Chile, in 2009. He is an Associate Professor with the Department of Computer Science, Universidad de Talca, Chile. In 2013, he carried out postdoctoral research with the Department of Computer Science, VU University Amsterdam, as part of his participation in the Linked Data Benchmark Council Project. He participates as a Researcher with the Millennium Institute for Foundational Research on Data, Chile. His research interests lie in the intersection of graph databases and the semantic web. Specifically, he works in the theory and design of graph query languages and the interoperability between RDF and graph databases.

• • •