Provenance-Based Routing in Probabilistic Graph Databases

Yann Ramusat
DI ENS, ENS, CNRS, PSL University
& Inria
Paris, France
yann.ramusat@ens.fr

Supervised by Pierre Senellart and Silviu Maniu

ABSTRACT

Optimizing routing queries over graphs is a rich research area with important applications, e.g., to road and transportation networks. Thanks to progress made during past decades, current-day systems are able to compute paths across cities in continent-sized areas, paths that are optimal in terms of distance or expected travel time. Nevertheless, the problem considered is quite limited, personal preferences cannot be handled effectively, and similar queries need to be computed separately. We explore a provenance-based framework as a way to extend the expressive power of routing queries, based on the idea of keeping track of meta-information about query results. This framework, useful to deal with such aspects as uncertainty or preferences, cannot always benefit of optimizations used for computing optimal routes, leading to impractical algorithms. The aim of our PhD is to improve on routing techniques based on provenance to apply them to real transportation networks.

1. CONTEXT

Progress made during past decades on optimization of routing (i.e., computation of optimal paths) in road and transportation networks led to current competing algorithms answering queries in milliseconds over continent-sized areas [4]. These algorithms exploit very specific properties of routing operations. Thus they tend to be very constrained and not able to handle the needs of real-life concerns, such as uncertainty about traffic congestion or personal preferences of the users.

To extend the expressive power of routing queries and to take into account this additional information, we propose using a framework based on *provenance annotations* over *graph databases*.

Graph databases are a common way to manage graph-like data, with applications to social network analysis, transportation networks, or the Semantic Web. A notable graph DBMS is Neo4j[1] and its Cypher query language[2]. These databases are typically queried using navigational queries, an abstraction for which are *Regular Path Queries* (RPQs) [3], which select pairs of vertices joined by a path whose label belongs to the regular language defined by the query.

In this PhD project, in order to incorporate additional information within a graph database, we enrich the graph with *provenance annotations*. These annotations are propagated to query results, and can be used to determine how the result has been computed and how it reacts to slight changes in the initial database. A mathematically rich way to do this is to choose provenance annotations to be elements of a *semiring* [12]. Semirings are well-suited to model computations (e.g., choices and sequences) carried along in computations such as the shortest-path problem, which can be rephrased as solving equational systems over semirings. This framework is expressive enough to deal with uncertainty and security clearances, among many other applications.

We thus use as a basis of our framework to compute enriched routing queries: graph databases, navigational queries, and semiring-based provenance.

Our past work [18] suggests that computing the semiring-based provenance of navigational queries over graph databases results in high complexity. It is strongly believed we cannot avoid a cubic (data)-complexity [22] for the general problem. To overcome this issue, we consider four main approaches to speed computations up.

- **Considering the shape of the network.** It has been shown in [15] that transportation networks exhibit a relatively low treewidth. Exploiting this low treewidth may help improve the running time of preprocessing or query evaluation. It is worth noting these ideas have already been applied to the routing problem [10] without provenance.
- **Restricting the expressive power of provenance annotations.** Considering subclasses of semirings can lead to more optimized algorithms, thus yielding a

[1] https://neo4j.com/
[2] http://www.opencypher.org/

Proceedings of the VLDB 2019 PhD Workshop, August 26th, 2019. Los Angeles, California.

trade-off between expressive power and cost of computation.

- **Linking to other models.** Previous work has considered optimizing the computation of provenance for recursive query languages such as Datalog [12, 9]; relating RPQs over graphs to these models may allow reusing these optimization techniques.

- **Adapting state-of-the-art routing algorithms.** Standard routing algorithms are orders of magnitude faster than provenance-aware routing algorithms because they rely on very specific properties of routing operations. It is worth determining whether they can be generalized for our purpose.

Ultimately our PhD aims to offer a strong theoretical foundation for future systems supporting non-trivial real-life routing applications. These four main directions need to be combined in an effective way to allow designers of such systems to apply the fastest algorithm possible without losing the capacity to handle the needs of their users.

The document is organized as follows: Section 2 discusses in more detail the state of the art in routing and provenance-aware routing algorithms. We then present in Section 3 some preliminary results based on some improvements of already known algorithms for specific classes of semirings (*bounded* semirings, *distributive lattices*, etc.). This leads us to Section 4 where we present a roadmap for the next two years and a half of our PhD research.

2. STATE OF THE ART

We will focus on two different areas of the research literature: one for the algorithms for the (simple) routing problem and one dedicated to algorithms for provenance-based routing in the context of our framework.

Routing algorithms. We provide an overview of the current techniques in routing we think ready to be generalized to our settings. Mostly, current competing algorithms rely either on the inherent hierarchy of the network (*hierarchical techniques*) or on the precomputation of distances between well-chosen pairs of vertices (*bounded-hop techniques*) [4].

Hierarchical techniques are based on the observation a subnetwork of important roads (such as highways) concentrate the traffic between sufficiently far away cities. This allows to scan few vertices for long distance queries. Two major algorithms belong to this framework: Contraction Hierarchies [11] and Reach [13].

On the other side, *bounded-hop techniques* such as Labeling algorithms [17] or Transit Nodes Routing [5, 6] permit to answer queries based on a virtual network composed of precomputed shortcuts and limiting to few-hop paths.

These two kinds of techniques can rely on each other, for example the choice for hops in the Labeling algorithm can be done based on the most important nodes discovered by the first step of a contraction hierarchy [4].

In [2] a new measure of the network (the *Highway dimension*) has been introduced. This measure provides a way to justify from a theoretical point of view the (sublinear) complexity of most of these techniques, dramatically helping our understanding of these techniques.

Provenance-aware routing algorithms. In [18] we generalized three existing graph algorithms to compute the provenance of regular path queries over graph databases. Each algorithm yields a different trade-off between time complexity and generality, as each requires different properties over the semiring.

Together, these algorithms already cover a large class of semirings used for provenance (**top-k**, **security**, etc.).

Experimental results suggest these approaches are complementary and practical for various kinds of provenance indications, even on a relatively large transport network.

In the following we do a brief review of them, their complexity and the situations were they can be applied.

- Dijkstra's algorithm can be successfully applied for computing single-source provenance. This algorithm can be applied whenever we have 0-*closed totally ordered* semirings (also known as *bounded* or *absorptive* semirings) [18]. Under these restrictions we can still compute **security clearances**.

- Mohri's algorithm [16] for the case when annotations belong to k-*closed* semirings. This algorithm is exponential in theory but experimental studies [16, 18] showed this algorithm is in fact practical in a real context. Under these restrictions we can compute for example **top-k shortest-paths** and **top-k distinct shortest-paths**. Note these restrictions are weaker as for Dijkstra so we can still apply this algorithm for **security clearances**.

- The last algorithm is inspired by the node elimination method to obtain the language recognized by a finite state automaton [8]. Notice that this problem can be expressed inside our framework using the *semiring of formal languages*. The provenance for one pair of nodes (x, y) is precisely the language recognized by the automaton with x as initial state and y as final state. We can then compute single-pair provenance **without any restriction** on the underlying semiring.

In introduction we were referring to semirings as being well-suited structures to model several problem classes in computer sciences. These problems can be unified using matrices over $*$-*semirings* (also known as *closed* semirings), semirings with a star operation. The theory of matrices over $*$-*semirings* [14, 1] exhibit number of similarities with linear algebra. All-pairs graph provenance is then equivalent to the computation of the *asteration* of the matrix corresponding to the graph representation with provenance tags as cell-values.

Notice the graph provenance is in fact an (infinite) sum over the provenances of all paths from the source node leading to the target node. Previous algorithms work over $*$-*semirings*. Infinite sums can only appear because of circles in our graphs, so we only need to be able to sum all the powers of a given values. To have a result semantically correct we need to ensure that $a^* = \sum_{n=0}^{\infty} a^n$ and that associativity and distributivity extends to these infinite sums. This class of semirings is commonly known as *countably complete star semirings*, or *c-complete star semirings*.

3. PRELIMINARY RESULTS

We now briefly present our preliminary results, achieved during the first months of our PhD research. We emphasize

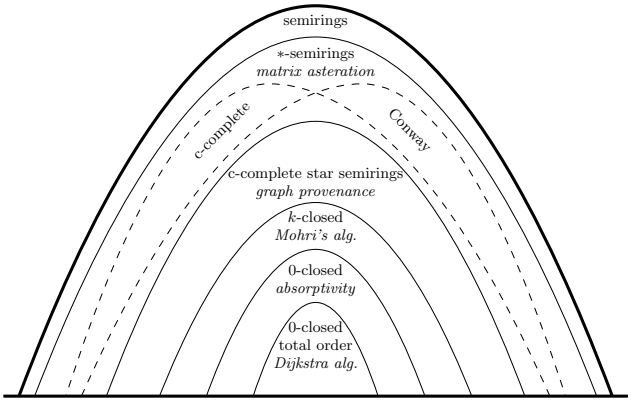

Figure 1: Taxonomy of the semirings we can use for graph provenance along with their key properties and/or associated algorithms.

how they allowed us to answer our research questions and how they can be classified according to the four main approaches discussed in the introduction.

A taxonomy of semirings. As a first task, to obtain a better understanding of the domain of semiring-based provenance, we classified all classes of semirings of interest, along with their key properties or the best-known algorithm for provenance-based routing in these semirings. We then produced a taxonomy of these semirings, that is graphically represented as Figure 1.

0-closed semirings. Our first concern was to overcome the need of a natural total order to ensure correctness of Dijkstra's algorithm. In [18], we gave an example where Dijkstra's algorithm fails when the semiring is 0-closed but not totally ordered: the semiring of natural numbers with greatest-common-divisor and least-common-multiple as semiring operators. For this class of 0-closed semirings, already a strong restriction, we do not know of any algorithm more efficient for the single-pair problem than either node elimination (cubic time) or Mohri's algorithm (exponential in theory, relatively efficient in practice). This is a huge gap with the situation of 0-closed totally ordered semirings for which Dijkstra's algorithm is polynomial-time. Our investigations led us to consider the case of 0-*closed multiplicatively idempotent semirings* (0-closed semirings in which multiplication is idempotent). It turns out these are equivalent to *bounded distributive lattices* [7]. A prominent example of such semiring in the database context is the **PosBool(X)** provenance semiring [12].

Relying on the rich mathematical theory behind lattices, we were able to design a parameterized algorithm based on Dijkstra's algorithm answering single-source provenance with a parameterized number of calls of Dijkstra's algorithm. We consider as a parameter the **width** of the chain decomposition of the *join-irreducible* elements of the lattice [20]. Intuitively, each element of the lattice can be uniquely represented as a combination of *join-irreducible* elements. Considering a *chain decomposition* of these elements allow for applying Dijkstra independently for each component of the product as they all are totally ordered.

THEOREM 1. *Let L be a* fixed *distributive lattice, with a chain decomposition of its join-irreducible elements of width w. Single-source provenance of an RPQ over a graph database of n nodes and m edges with annotations in L can be computed using w applications of Dijkstra's algorithm. This results in a complexity for the whole computation of $\mathcal{O}(w \times (m + n \log n))$, assuming semiring operations in L take constant time.*

Links with Datalog queries. Recently, it has been shown we can make use of circuits to represent provenance for Datalog queries in a much more efficient way [9]. It is worth noting that, in this framework, *distributive lattices* and 0-*closed* semirings are also distinguished classes of semirings for optimizing queries; **PosBool(X)** and **Sorp(X)** being respectively the *free distributive lattice* and the *free* 0-*closed semiring*.

In order to permit investigations for the links between the two instances of the provenance concept, we describe how to translate our database and our query into a Datalog query.

Graph database. Given a graph database $G = (V, E, w)$ we can encode it into an edb. Let Σ be the alphabet for edge labels, create $|\Sigma|$ binary relations $\{R_\sigma \mid \sigma \in \Sigma\}$ and populate them with facts corresponding to existing labeled edges: for each $e = (u, v) \in E$, fact $R_{w(e)}(u, v)$ holds. We tag these facts with the same provenance indication as for the edges.

RPQ. We recursively convert an RPQ L into a (linear) Datalog query:

- If $L = a$, $R_L(x, y) \leftarrow R_a(x, y)$,
- If $L = L_1 \cup L_2$, $R_L(x, y) \leftarrow R_{L_1}(x, y)$ and $R_L(x, y) \leftarrow R_{L_2}(x, y)$,
- If $L = L_1 \cdot L_2$, $R_L(x, z) \leftarrow R_{L_1}(x, y), R_{L_2}(y, z)$,
- If $L = L_1^*$, $R_L(x, z) \leftarrow R_L(x, y), R_{L_1}(y, z)$ and $R_L(x, x) \leftarrow$.

The size of the resulting (idb) program is linear in the size of the RPQ and linear (for the edb predicates) in the database instance.

4. ROADMAP

Our PhD research started at the beginning of September 2018 and is expected to last three years. As such, we are still at a very preliminary stage of our research.

Our initial observations opened new research directions we would like to investigate.

Comparison with Datalog queries. After observing we can translate RPQs against graph databases into Datalog programs over relational encodings of these graphs, we would now like to compare the expressive power and efficiency of algorithms for computing the provenance of RPQs on graphs to that for computing provenance of Datalog queries over relational databases. We are also interested in investigating a *reverse* translation, to determine which fragment of Datalog can be translated back into our model, in order to derive complexity bounds and obtain insights on the expressive power of our framework.

Another to way to benefit from this observation is to compare actual optimizations for Datalog queries based on the analysis of derivation trees [9] with ours: Dijkstra's algorithm and extended Dijkstra's algorithm for *distributive lattices*.

Adapting current state-of-the-art routing algorithms. We want to investigate how we can adapt routing algorithms introduced in Section 2 within our framework. One major challenge is that these routing algorithms rely on the assumption that a small number of nodes or junctions concentrate most of the long-distance traffic. This cannot be applied to the computation of provenance in arbitrary semirings: for example, in the counting semiring, we need to be able to count all paths between two nodes, which seems to require to explore all these paths. A natural question is then: does there exist specific semirings for which the assumption holds, and for which we can apply these algorithms?

Lower bounds. Finally, we would like to address the problem of finding lower complexity bounds for our general framework, especially involving c-complete star semirings (resp., $*$-semirings). For now, the complexities of the algorithms we can use for all-pairs, single-source, and single-pair provenance are the same. Thus we would like to know if computing the provenance for one pair is indeed as difficult as computing the full matrix asteration. These bounds are commonly hard to prove but we could rely on already known (or suspected) bounds for the APP problem [21] or on some results based on circuit complexity (such as Theorem 1 of [9]. As our background is not in (circuit)-complexity theory but in routing algorithms and database theory, this may require a collaboration with specialists in this area.

5. CONCLUSION

We gave an overview of a provenance-based framework for routing queries, and of different approaches considered to lower the complexity of query evaluation: considering the shape of the network, restricting the expressive power of provenance annotations, linking to other models, and adapting state-of-the-art routing algorithms.

Our research has so far involved considering a new class of semirings together with an effective algorithm. We have observed similarities with semirings for which optimizations of Datalog provenance computation have been proposed, leading us to consider translation of our queries into Datalog programs for further investigations.

We then have exposed intended directions for our PhD work, pursuing our current research, and finally combining them to obtain theoretical and practical results applicable to real-world transportation networks.

As the aim of our PhD is to offer a strong theoretical foundation for an eventual implementation of a provenance aware query optimizer/processor in a graph database system, we conclude with a final note concerning how to adapt such an existing system to support provenance. One way to proceed is to dynamically rewrite queries to make them process auxiliary data carrying provenance information; this idea has been successfully applied in the context of a relational database system [19]. The major drawback of this approach is that most of the optimization strategies are no longer applicable, thus leading to a non-negligible computation time overhead; this would be the bulk of our system-focused research.

6. REFERENCES

[1] S. K. Abdali. Parallel computations in *-semirings. *Computational Algebra*, pages 1–16, 1994.

[2] I. Abraham, A. Fiat, A. V. Goldberg, and R. F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *SODA*, pages 782–793. 2010.

[3] P. Barceló Baeza. Querying graph databases. In *PODS*, pages 175–188, 2013.

[4] H. Bast, D. Delling, A. V. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck. Route planning in transportation networks. *CoRR*, abs/1504.05140, 2015.

[5] H. Bast, S. Funke, and D. Matijević. Ultrafast shortest-path queries via transit nodes. In *The shortest path problem: Ninth DIMACS implementation challenge*, pages 175–192. 2006.

[6] H. Bast, S. Funke, P. Sanders, and D. Schultes. Fast routing in road networks with transit nodes. *Science*, 316:566, 2007.

[7] S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint satisfaction and optimization. *JACM*, 44(2):201–236, 1997.

[8] J. A. Brzozowski and E. J. McCluskey. Signal flow graph techniques for sequential circuit state diagrams. *IEEE Trans. Electronic Computers*, EC-12(2):67–76, 1963.

[9] D. Deutch, T. Milo, S. Roy, and V. Tannen. Circuits for Datalog Provenance. In *ICDT*, pages 201–212, 2014.

[10] R. Geisberger, P. Sanders, D. Schultes, and D. Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In *Experimental Algorithms*, pages 319–333, 2008.

[11] R. Geisberger, P. Sanders, D. Schultes, and C. Vetter. Exact routing in large road networks using contraction hierarchies. *Transportation Science*, 46(3):388–404, 2012.

[12] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS*, pages 31–40, 2007.

[13] R. Gutman. Reach-based routing: A new approach to shortest path algorithms optimized for road networks. In *ALENEX*, pages 100–111, 2004.

[14] D. J. Lehmann. Algebraic structures for transitive closure. *TCS*, 4(1):59–76, 1977.

[15] S. Maniu, P. Senellart, and S. Jog. An experimental study of the treewidth of real-world graph data. In *ICDT*, pages 12:1–12:18, 2019.

[16] M. Mohri. Semiring frameworks and algorithms for shortest-distance problems. *J. Autom. Lang. Comb.*, 7(3):321–350, 2002.

[17] D. Peleg. Proximity-preserving labeling schemes. *J. Graph Theory*, 33(3):167–176, 2000.

[18] Y. Ramusat, S. Maniu, and P. Senellart. Semiring provenance over graph databases. In *TaPP*, 2018.

[19] P. Senellart, L. Jachiet, S. Maniu, and Y. Ramusat. Provsql: Provenance and probability management in postgresql. In *Proc. VLDB*, pages 2034–2037, Rio de Janeiro, Brazil, Aug. 2018. Demonstration.

[20] M. Siggers. On the representation of finite distributive lattices. *arXiv [math]*, abs/1412.0011, 2014.

[21] R. Williams. Faster all-pairs shortest paths via circuit complexity. *CoRR*, abs/1312.6680, 2013.

[22] R. Williams. Faster all-pairs shortest paths via circuit complexity. In *STOC*, pages 664–673, 2014.