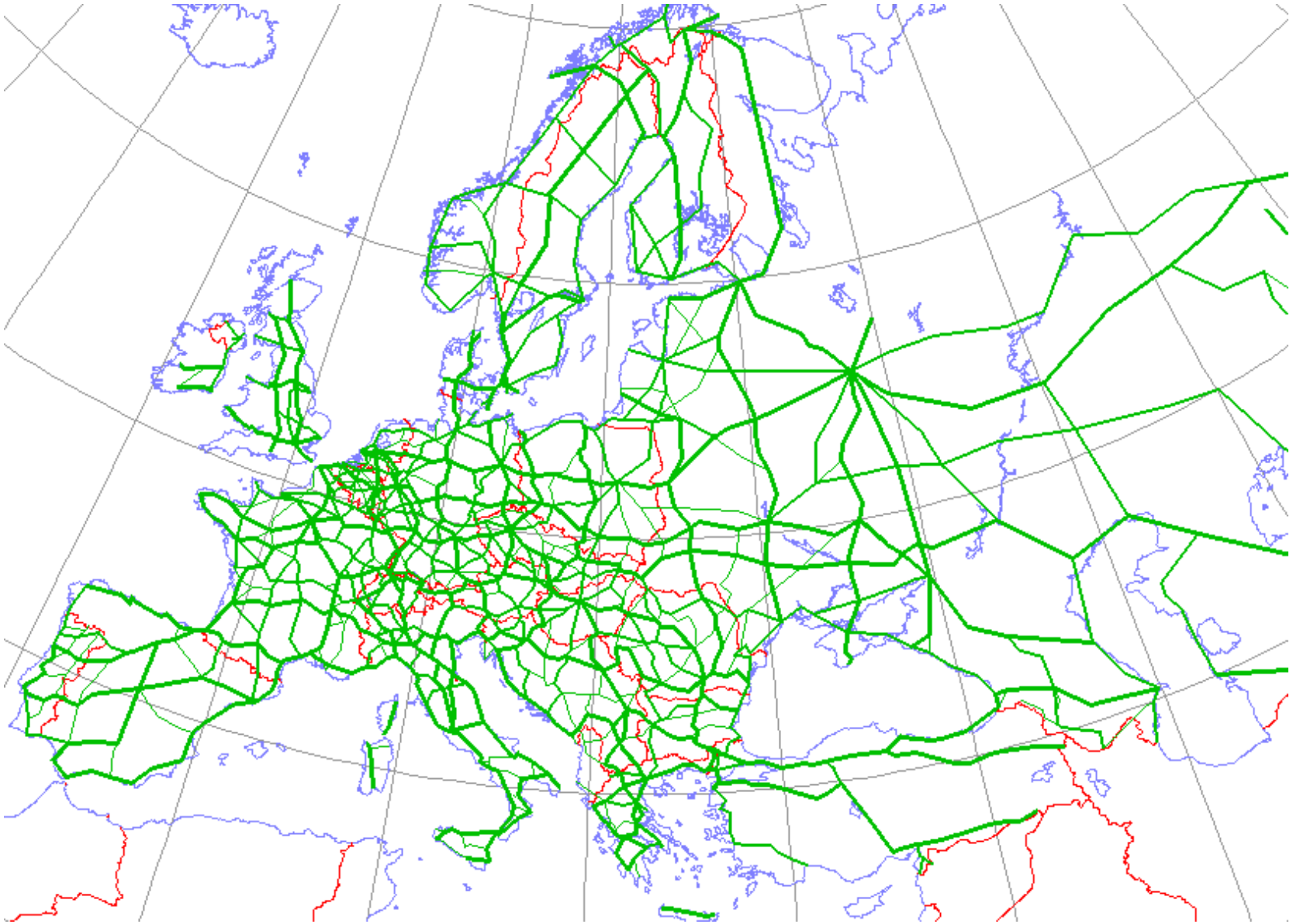
Provenance-Based Algorithms for Rich Queries over Graph Databases

BDA'20

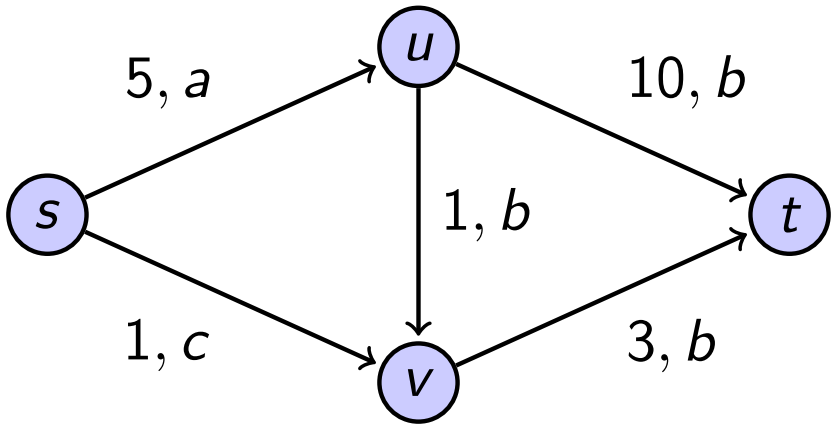
October 29, 2020

Silviu Maniu, **Yann Ramusat**, Pierre Senellart



https://en.wikipedia.org/wiki/International_E-road_network

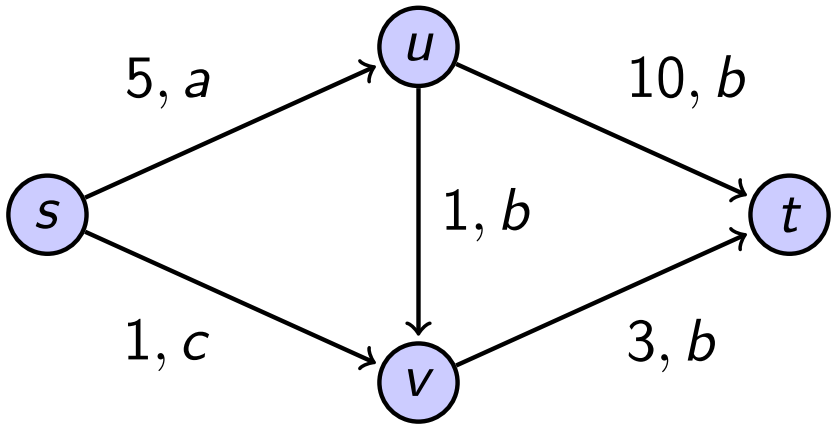
Working example (Tropical semiring)



These integers represent time to move between two vertices.

What is the minimum travel time between s and t?

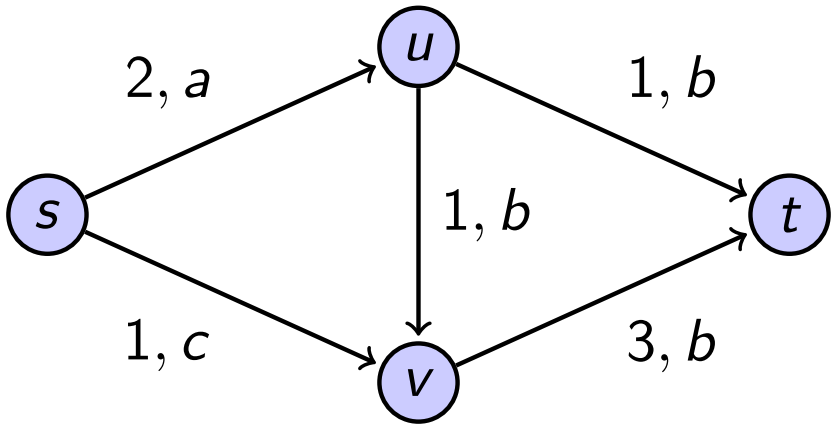
Working example (Tropical semiring)



These integers represent time to move between two vertices.
What is the minimum travel time between s and t?

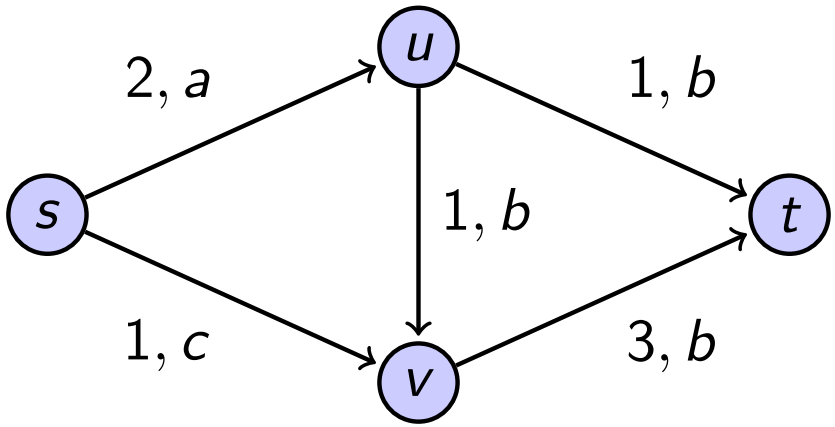
And now, if we only want to consider paths going through two b edges?

Working example (Counting semiring)



These integers represent number of paths between two vertices.
What is the total number of paths between s and t?

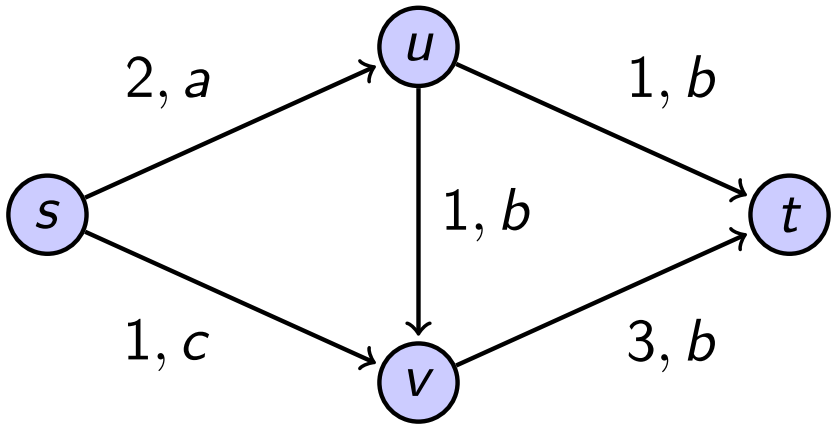
Working example (Counting semiring)



These integers represent number of paths between two vertices.
What is the total number of paths between s and t?

And now, if we only want to consider paths going through two b edges?

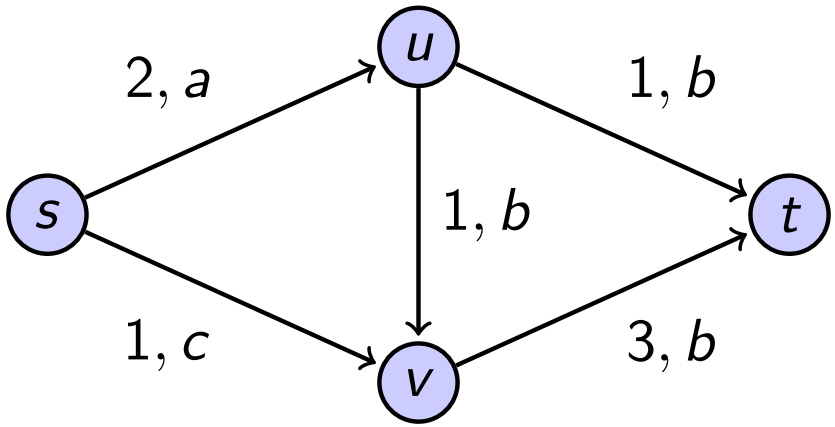
Working example (Top-2 semiring)



These integers represent time to move between two vertices.

What is the best two travel times between s and t?

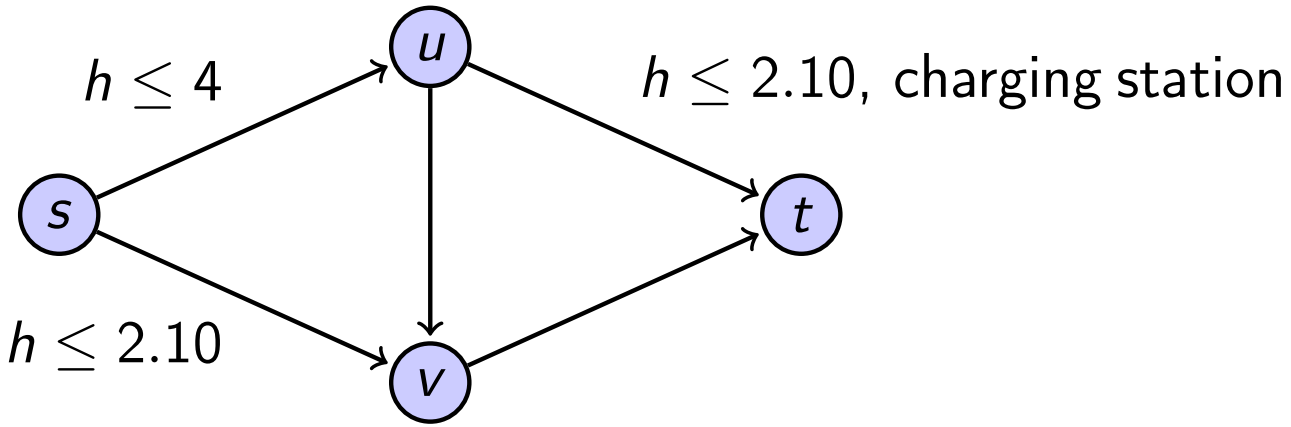
Working example (Top-2 semiring)



These integers represent time to move between two vertices.
What is the best two travel times between s and t?

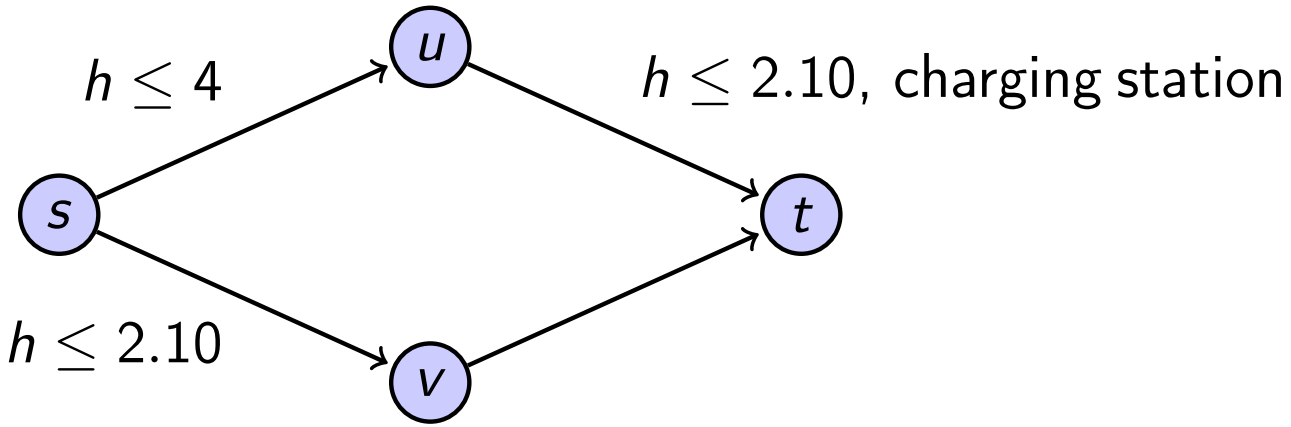
And now, if we only want to consider paths going through two b edges?

Working example (k-feature semiring)



There exists a path from s to t going through a charging station.
And there exists another one permitting $3m$ vehicles to reach t from s.

Working example (k-feature semiring)



There exists a path from s to t going through a charging station.

Contents

Graph Databases

Graph databases are a common way to manage graph-like data.

Notable applications to social network analysis, transportation networks, or the Semantic Web.

Typically queried using navigational queries.

We will focus on Regular Path Queries (RPQs) (Barceló Baeza, 2013): selecting pairs of vertices joined by a path whose label belongs to the regular language defined by the query.

Provenance annotations

To incorporate additional information within a graph database to address our problem, 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,
- how it reacts to slight changes in the initial database.

A strong mathematical foundation is to choose provenance annotations to be elements of a semiring (Green et al., 2007).

Semiring-based provenance

Semirings are a well-suited model for operations (e.g., choices and sequences) carried along in computations.

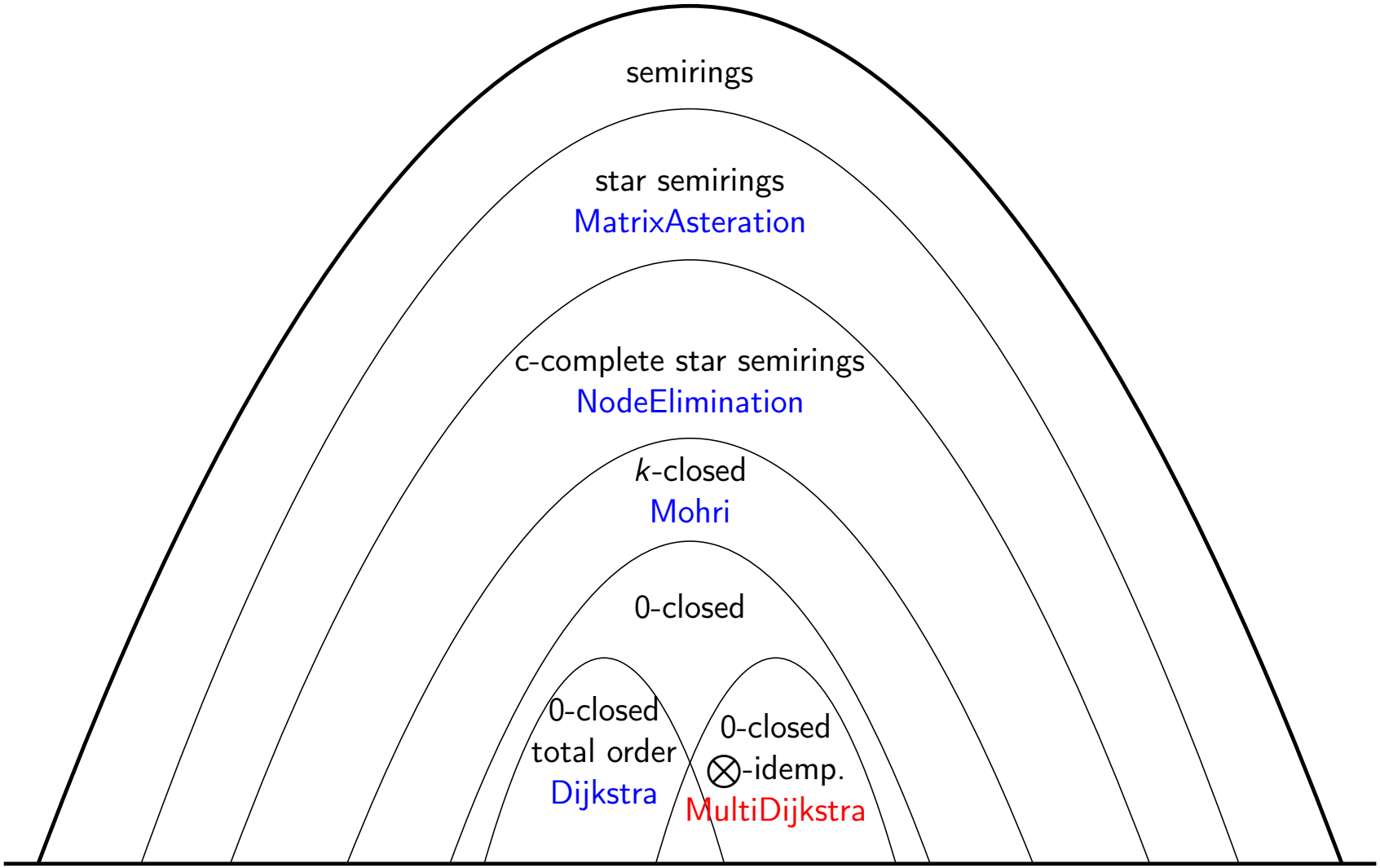
The shortest-path problem can be rephrased as solving equational systems over semirings.

Semirings are expressive enough to deal with uncertainty and access restrictions, among many other applications.

Also, they can be used to model existential and universal quantifiers over paths.

Contents

Hierarchy


semirings
star semirings
MatrixAsteration
c-complete star semirings
NodeElimination
k-closed
Mohri
0-closed
0-closed
total order
Dijkstra
0-closed
⊗-idemp.
MultiDijkstra

Complexities

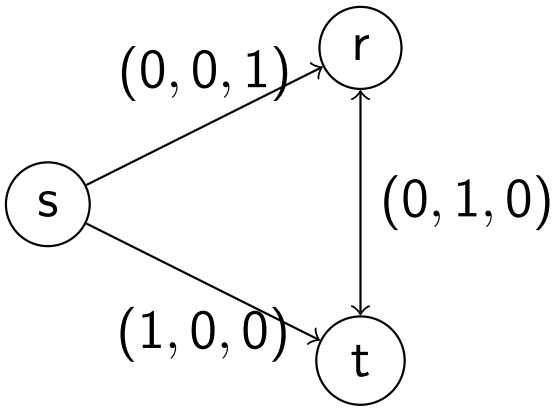
Name	**Semiring property**	**Time complexity (with provenance)**
MatrixAsteration	star	$\mathcal{O}(\lvert V\rvert T_* + \lvert V\rvert^3(T_\oplus + T_\otimes))$
NodeElimination	c-complete star	$\mathcal{O}(\lvert V\rvert T_* + \lvert V\rvert^3(T_\oplus + T_\otimes))$
Mohri	k-closed	Exponential
MultiDijkstra	0-closed $\otimes$-idempotent	$\mathcal{O}\left(\ell \times (\mathrm{T}_\oplus \lvert V\rvert \log \lvert V\rvert + \lvert E\rvert (T_\oplus + T_\otimes))\right)$
Dijkstra	0-closed total ordered	$\mathcal{O}(\mathrm{T}_\oplus \lvert V\rvert \log \lvert V\rvert + \lvert E\rvert (T_\oplus + T_\otimes))$

star: no restriction,
c-complete star: number of paths,
k-closed: top-k (distinct) shortest-distances,
0-closed $\otimes$-idempotent: k-features,
0-closed total ordered: access restrictions, shortest-path.

Contents

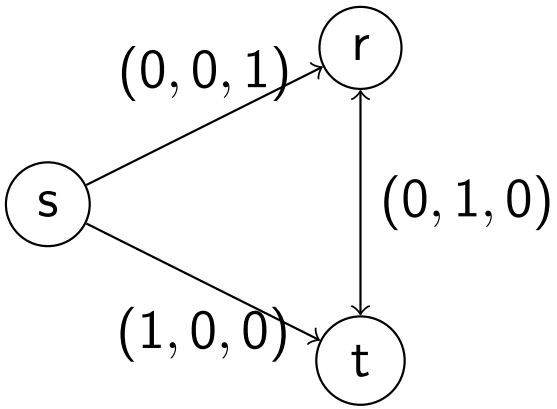
Counter example to Dijkstra

Using the 3-feature semiring $(\{0,1\}^3, \min, \max, (1,1,1), (0,0,0))$



Counter example to Dijkstra

Using the 3-feature semiring $(\{0,1\}^3, \min, \max, (1,1,1), (0,0,0))$

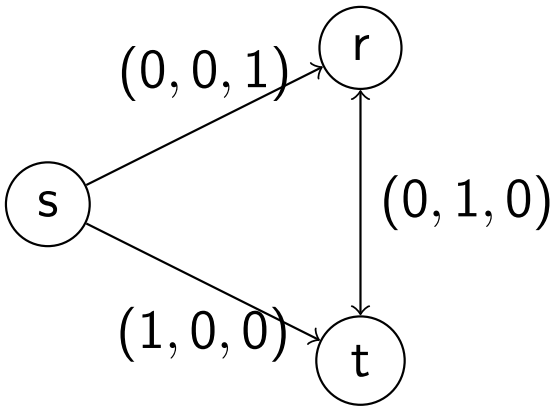


provenance between s and t is:

$$\min\left(\max\left((0,0,1),(0,1,0)\right),(1,0,0)\right) = (0,0,0)$$

Counter example to Dijkstra

Using the 3-feature semiring $(\{0,1\}^3, \min, \max, (1,1,1), (0,0,0))$



provenance between s and t is:

$$\min\left(\max\left((0,0,1),(0,1,0)\right),(1,0,0)\right) = (0,0,0)$$

provenance between s and r is:

$$\min\left(\max\left((1,0,0),(0,1,0)\right),(0,0,1)\right) = (0,0,0)$$

0-closed multiplicatively idempotent semirings

Equivalent to bounded distributive lattices.

A prominent example of such semiring in the database context is the PosBool(X) provenance semiring (Green et al., 2007).

Idea: we consider as a parameter the width (ℓ) of the chain decomposition of the join-irreducible elements of the lattice (Siggers, 2014).

Each element of the lattice can be uniquely represented as a combination of join-irreducible elements.

Dijkstra can be applied independently for each component of the product as they all are totally ordered.

MultiDijkstra

Algorithm 1 MultiDijkstra – single-pair

Require: $(G = (V, E, w), s, t)$ a graph database with provenance indication over S, the source s and the target t.

Ensure: Single-pair provenance of the reachability query from s to t.

1: **for** each edge $e \in E$ **do**
2: $\quad decompose(w(e))$
3: **end for**
4: **for** each dimension i **do**
5: $\quad d_i \leftarrow Dijkstra(s, t, i)$
6: **end for**
7: **return** $recompose(d_1, \ldots, d_n)$

Complexity: $\mathcal{O}(\ell \cdot (m + n \log n))$.

Contents

Methodology and datasets

type	name	# of vertices	# of edges	tw
infrastructure	Paris	4 325 486	5 395 531	55–521
	Stif	17 720	31 799	28–86
	USPowerGrid	4 941	6 594	10–18
	Rome99	3 353	4 831	5–50
social	Facebook	4 039	88 234	142–237
biology	Yeast	2 284	6 646	54–255

Only Rome99 has initial weights (over TropicalSemiring).

For the others, weights are uniformly generated between 1 and 3 000.

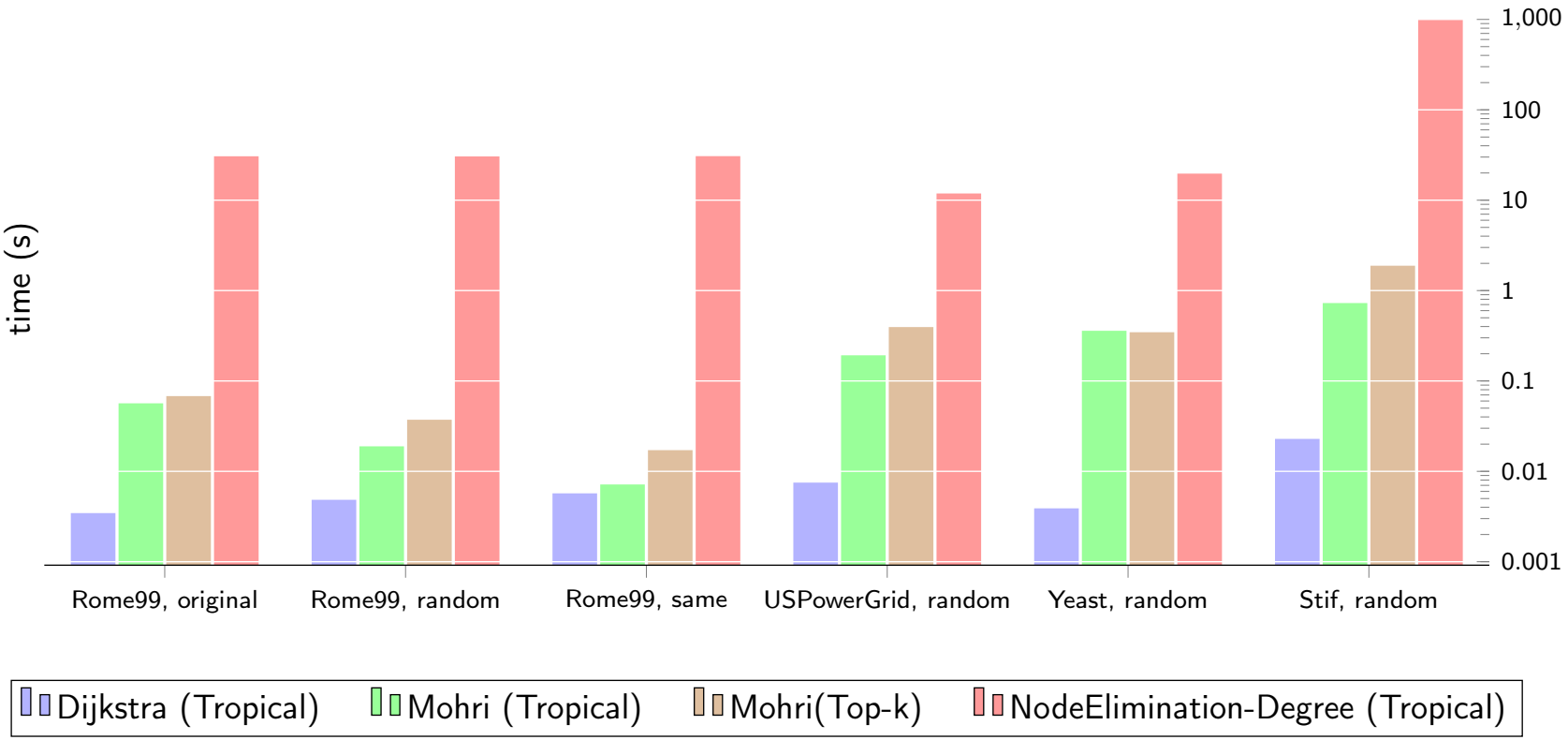
Also use constant-weight setting, where all weights equal to 1.

Average over 10 runs (random choices of origin and destination vertices).

Datasets from `https://github.com/smaniu/treewidth/`

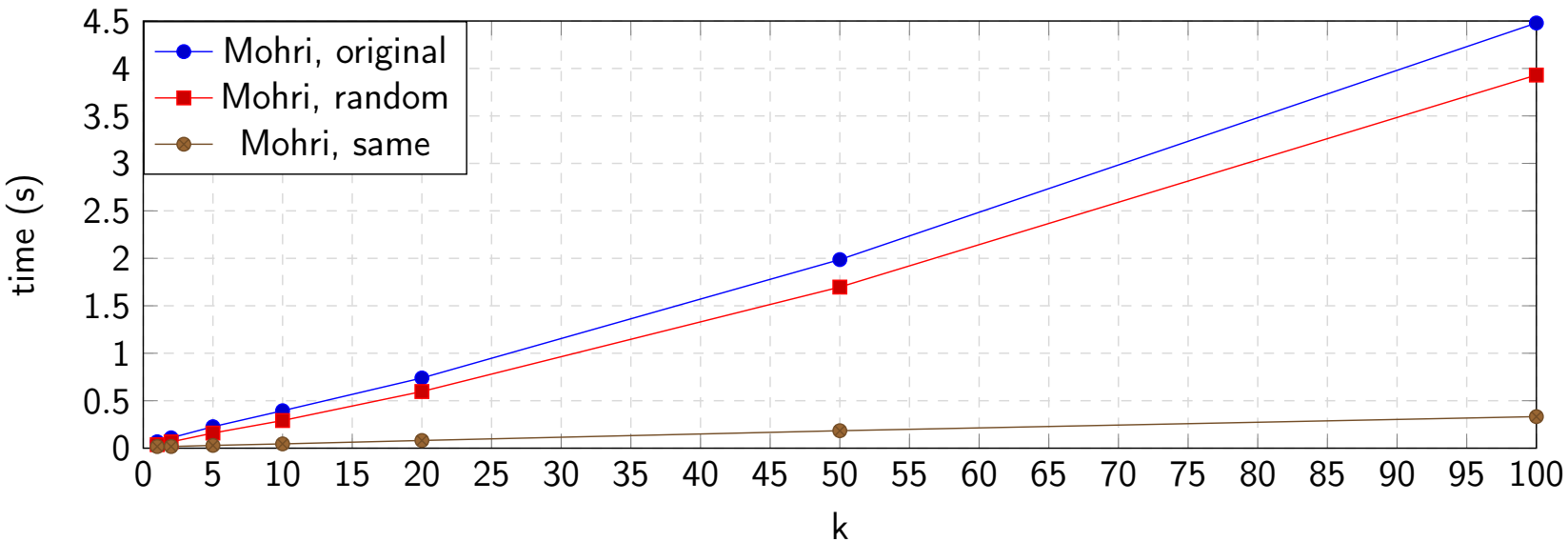
Evaluating shortest distances

Comparison between all of our algorithms for the computation of the shortest-distance



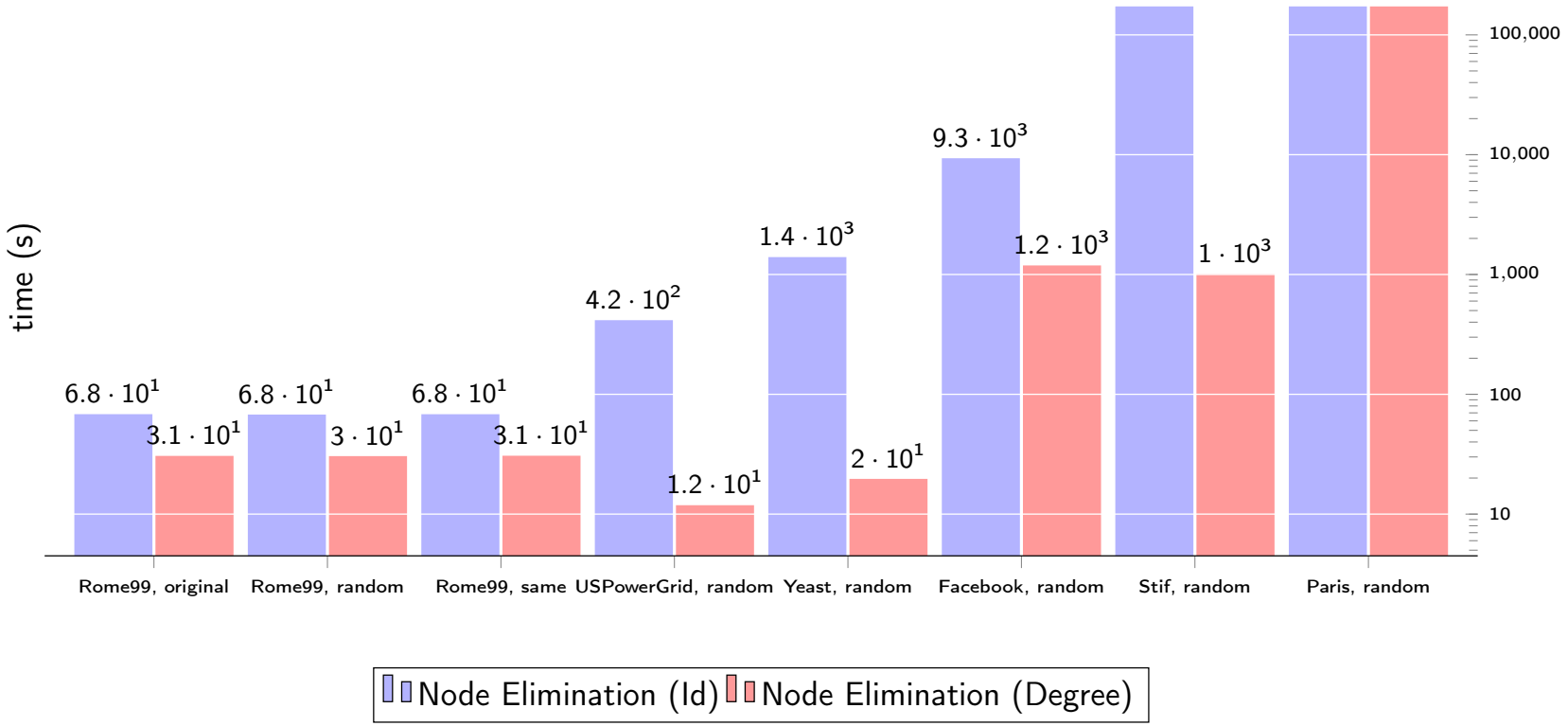
Mohri in practice

Running time and the number of relaxation steps performed by Mohri's algorithm



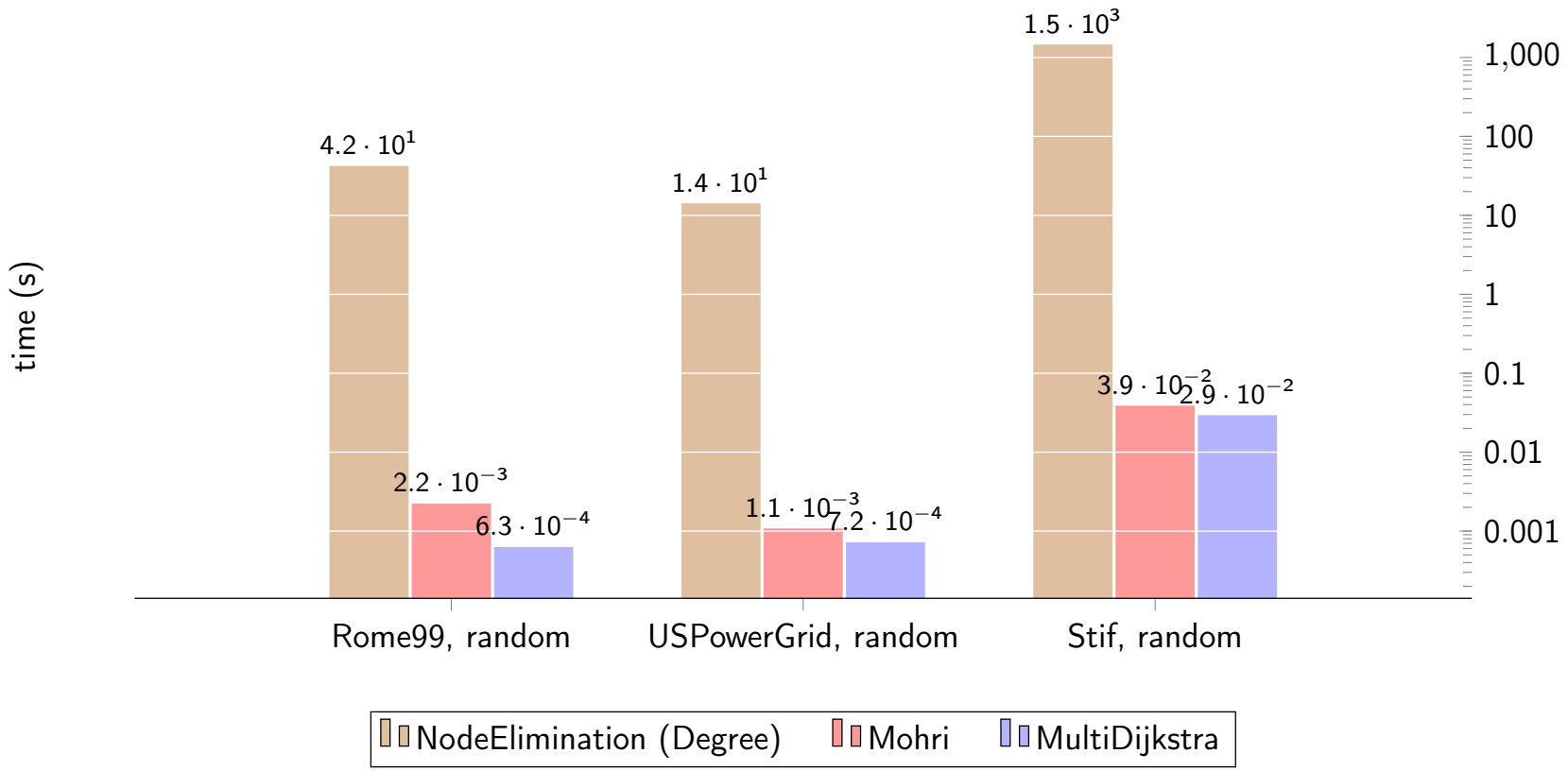
Ordering for NodeElimination

Impact of the elimination order on the overall performance for the Node Elimination algorithm over all the datasets



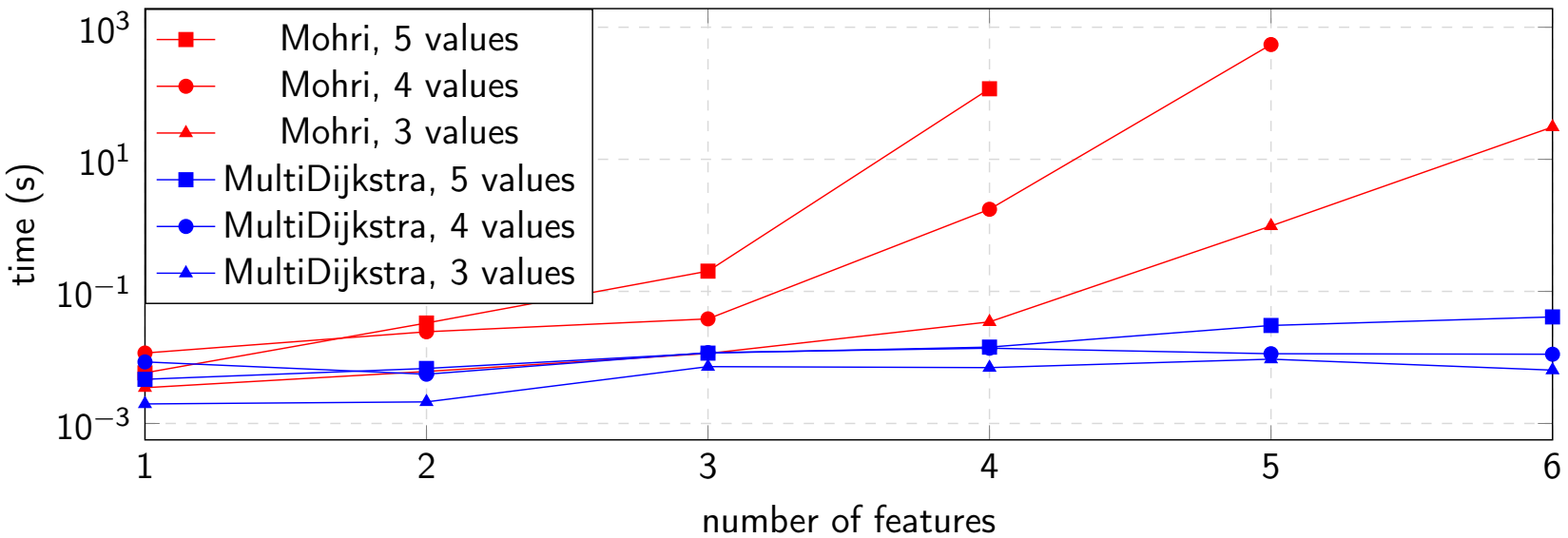
MultiDijkstra

Comparison between NodeElimination, Mohri, and MultiDijkstra (3-feature semiring)



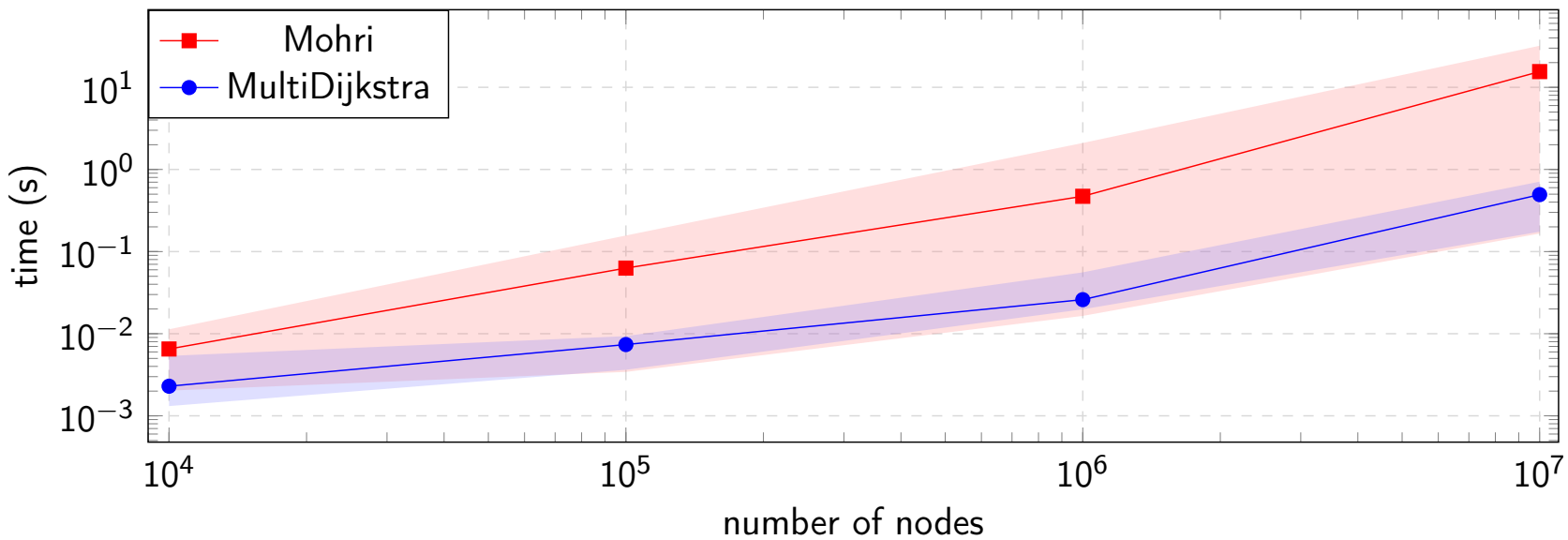
MultiDijkstra

Computation time for Mohri and MultiDijkstra depending on the number of dimensions and their sizes (Rome99, k-feature semiring)



MultiDijkstra

Average computation time for Mohri and MultiDijkstra over random graphs depending on the number of nodes; shaded areas indicate minimum and maximum computation times observed (3-feature semiring)



Contents

Perspectives

Compare expressive power and efficiency with Datalog provenance.

Optimization techniques such as Dijkstra's like algorithms can be applied for hypergraphs and context-free grammars (very close to Datalog).

Striking similarities between important classes of semiring in most of these frameworks.

Bibliography I

Barceló Baeza, P. (2013). Querying graph databases. In *PODS*, pages 175–188.

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

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