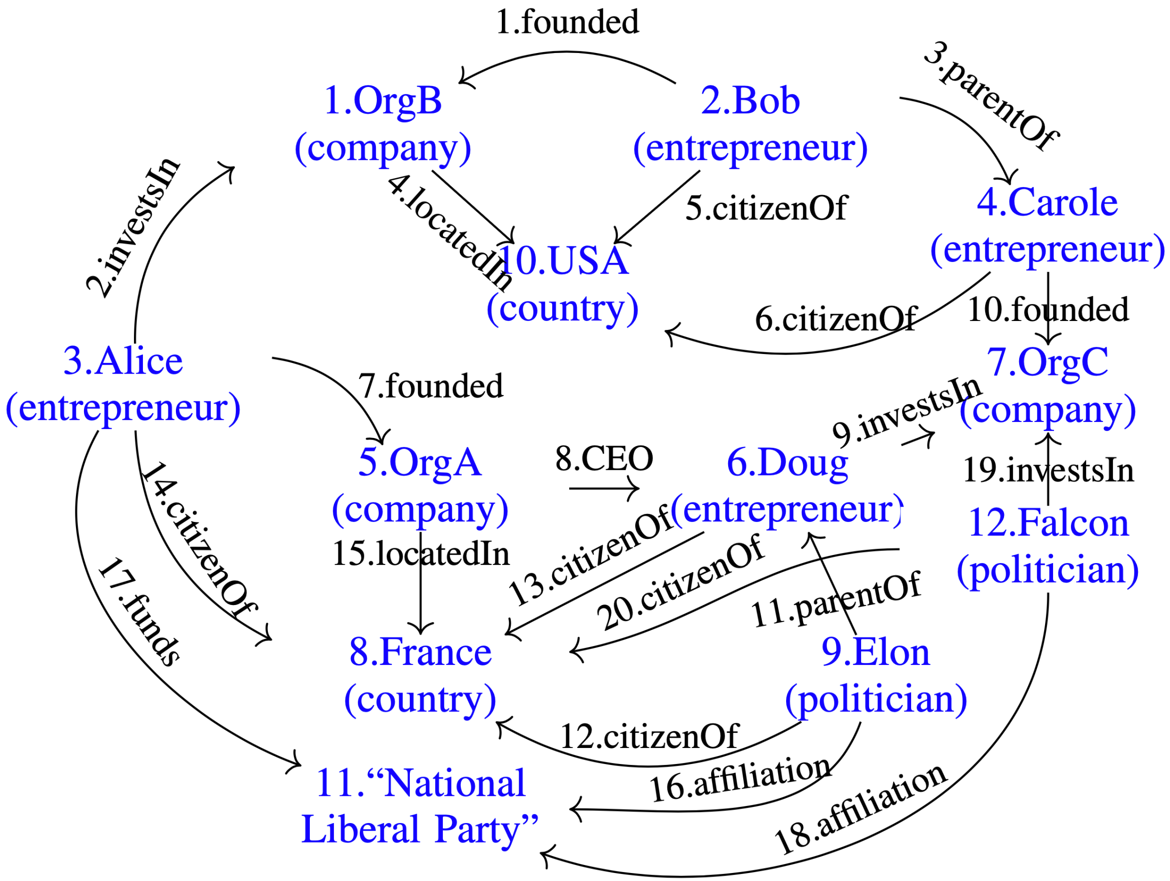
INTEGRATING CONNECTION SEARCH IN GRAPH QUERIES

Angelos C. Anadiotis, Ioana Manolescu, Madhulika Mohanty

Inria & IPP

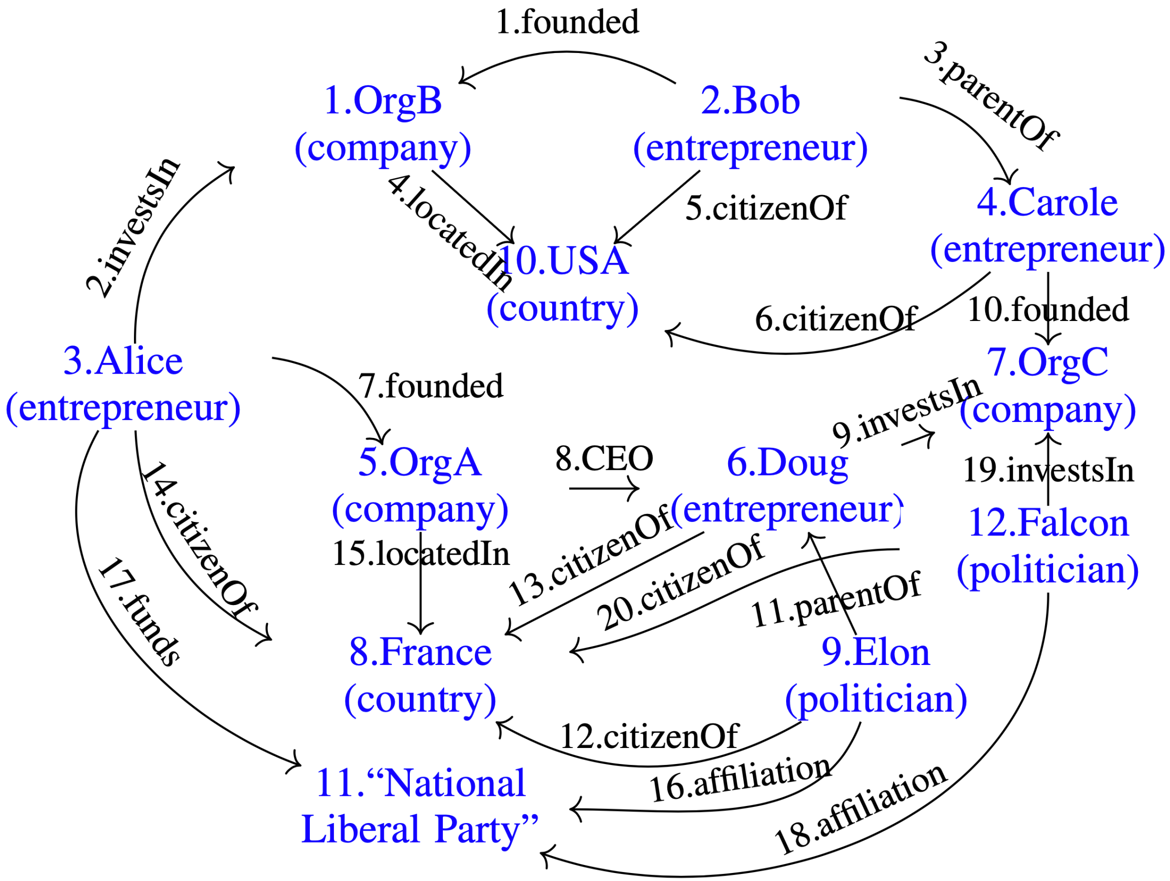
ICDE2023
Anaheim, CA April 3 – 7, 2023

MOTIVATING EXAMPLE



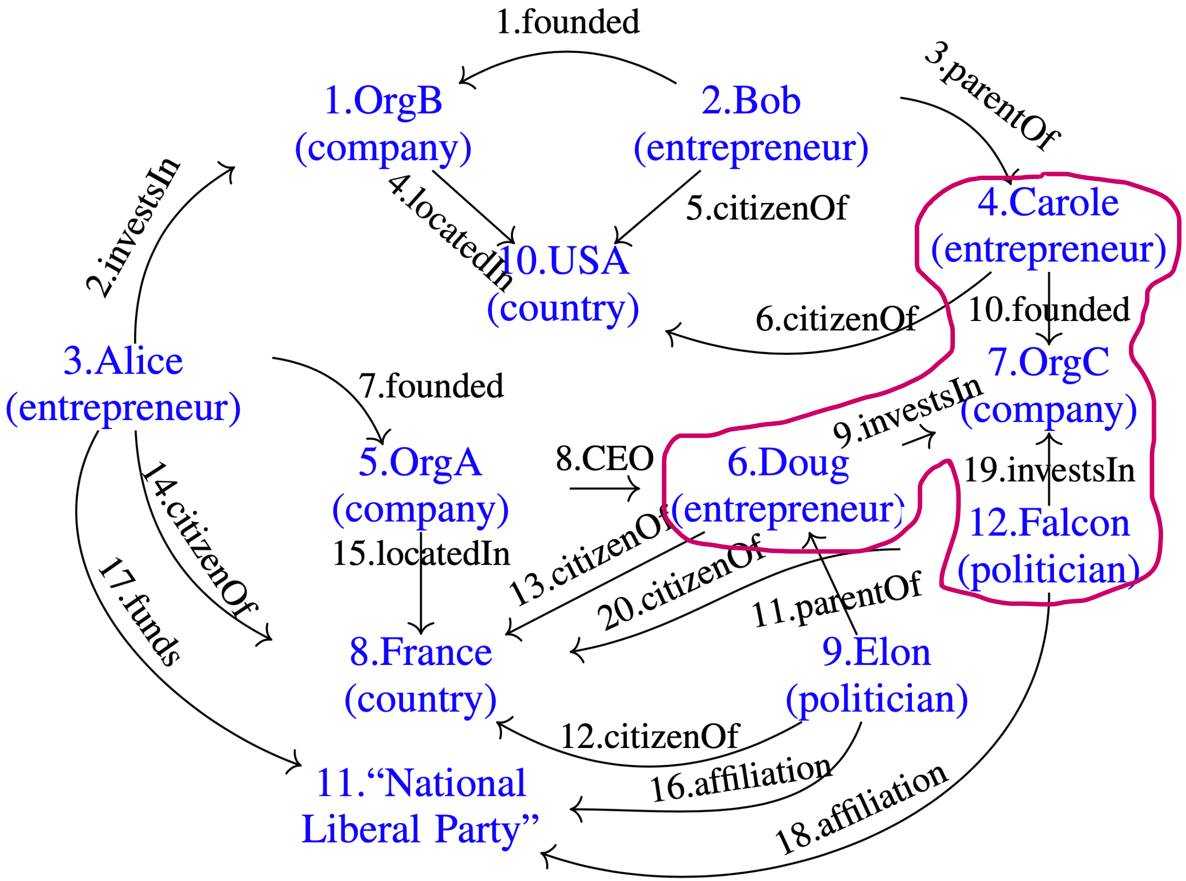
MOTIVATING EXAMPLE

How are the US entrepreneurs, French entrepreneurs and French politicians related?



MOTIVATING EXAMPLE

How are the US entrepreneurs, French entrepreneurs and French politicians related?



Connecting trees.

Doug, Falcon and Carole are leaves.

STATE OF THE ART

Requirements/ Existing	Exact match	(Label-constrained) regular paths between any two nodes	Connecting tree search
SPARQL	✓ (US entrepreneurs, French politicians, French entrepreneurs)	✓ (check but not return) $x – knows*-> $y	✗
Cypher/GQL	✓	✓ $x-[*]-$y	✗
Keyword Search Algorithms (BANKS, BLINKS, DBXplorer, etc.)	✗	✗	✓ (based on keywords alone, various pruning strategies)

How are the US entrepreneurs, French entrepreneurs and French politicians related?

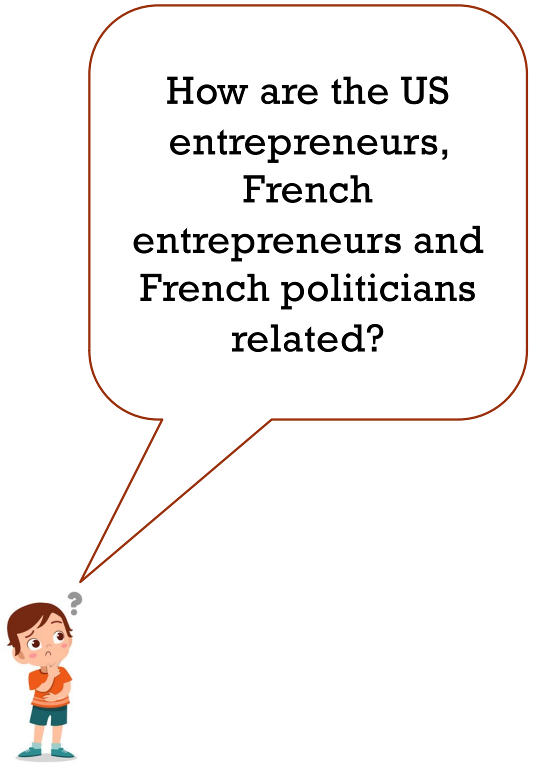
STATE OF THE ART

Requirements/ Existing	Exact match	(Label-constrained) regular paths between any two nodes	Connecting tree search
SPARQL	✓ (US entrepreneurs, French politicians, French entrepreneurs)	✓ (check but not return) $x – knows*-> $y	✗
Cypher/GQL	✓	✓ $x-[*]-$y	✗
Keyword Search Algorithms (BANKS, BLINKS, DBXplorer, etc.)	✗	✗	✓ (based on keywords alone, various pruning strategies)

Keyword Search has high complexity (Group Steiner Tree – NP-Hard)

How are the US entrepreneurs, French entrepreneurs and French politicians related?

REQUIREMENTS



(P1) A **query language** supporting such queries

REQUIREMENTS

How are the US entrepreneurs, French entrepreneurs and French politicians related?

(P1) A **query language** supporting such queries

(P2) General tree search

- Undirected search.
- Find all answers (under space and time budget).
- Independent of the cost function.

REQUIREMENTS

How are the US entrepreneurs, French entrepreneurs and French politicians related?

(P1) A **query language** supporting such queries

(P2) General tree search

- Undirected search.
- Find all answers (under space and time budget).
- Independent of the cost function.

(P3) Efficient execution algorithms

SUPPORT FOR EXTENDED QUERIES (EQ)

Given a graph and set of node/edge properties, GPML supports:

- **Path Patterns (PPs)** of the form:

  MATCH

  (v: Alice WHERE v.type=entrepreneur)

  −[e: citizenOf]→

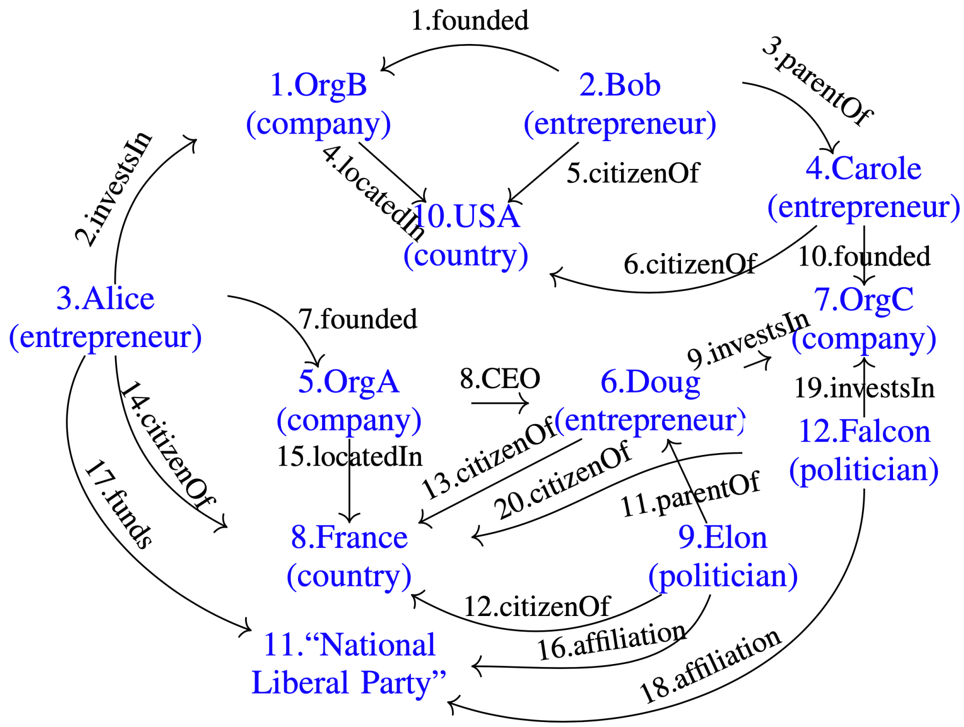
  (w WHERE w.type=country)

- PPs can contain Regular Paths:

  MATCH p = (x) −[y: founded]→∗(z)

- **Graph Patterns (GPs)**

  Conjunction of PPs



SUPPORT FOR EXTENDED QUERIES (EQ)

Given a graph and set of node/edge properties, GPML supports:

- ➢ **Path Patterns (PPs)** of the form:

  MATCH

  (v: Alice WHERE v.type=entrepreneur)

  −[e: citizenOf]→

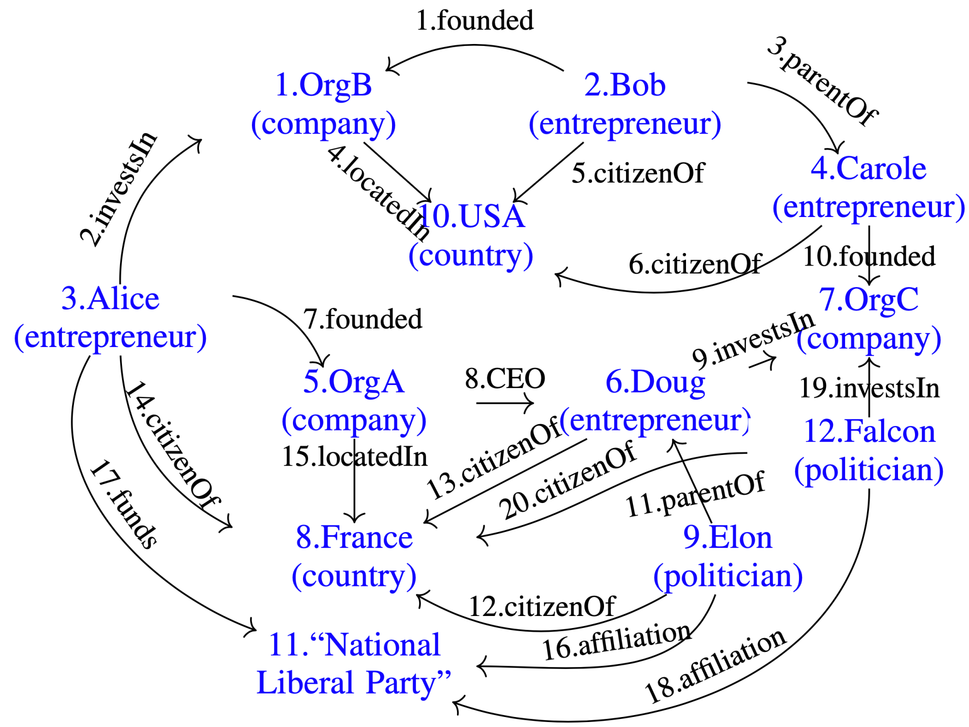
  (w WHERE w.type=country)

- ➢ PPs can contain Regular Paths:

  MATCH p = (x) −[y: founded]→∗(z)

- ➢ **Graph Patterns (GPs)**

  Conjunction of PPs



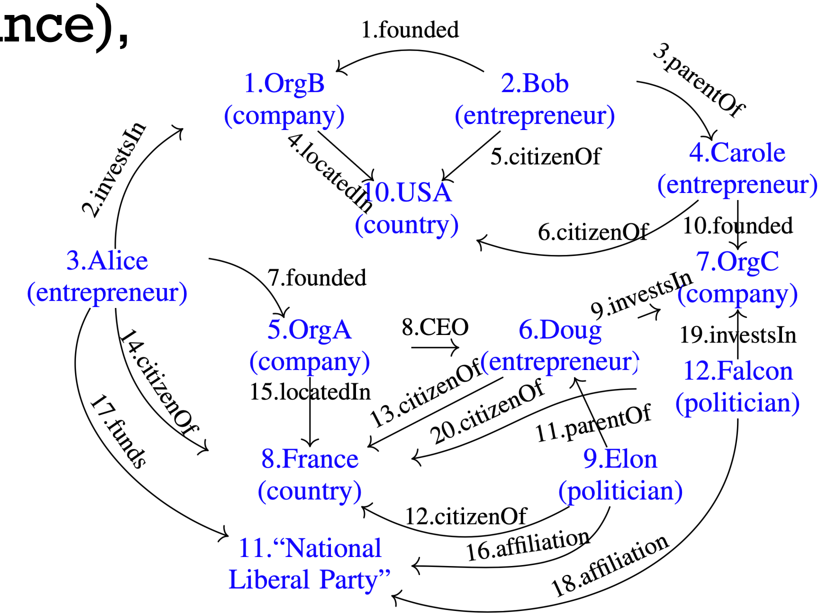
Our extension: **Connecting Tree Patterns (CTPs)**

- *n* input variables, 1 output variable (x, y, z, w)

EXAMPLE QUERY

How are the US entrepreneurs, French entrepreneurs and French politicians related?

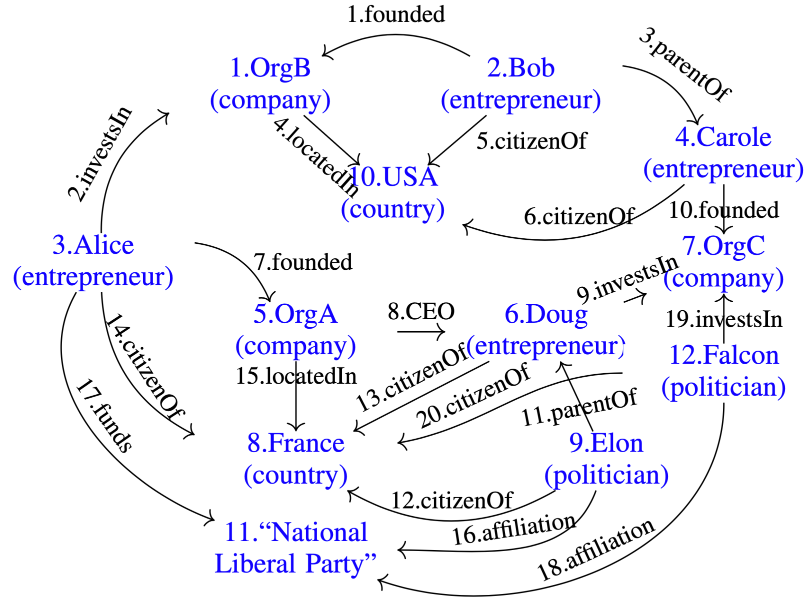
MATCH
(x WHERE x.type = entrepreneur) −[a: citizenOf]→ (b: USA),
(y WHERE y.type= entrepreneur) −[c: citizenOf]→ (d: France),
(z WHERE z.type = politician) −[e: citizenOf]→ (f: France),
(x, y, z, w)
RETURN w;



QUERY SEMANTICS (GP)

MATCH
(x WHERE x.type = entrepreneur) −[a: citizenOf]→ (b: USA),
(y WHERE y.type= entrepreneur) −[c: citizenOf]→ (d: France),
(z WHERE z.type = politician) −[e: citizenOf]→ (f: France),
(x, y, z, w)
RETURN w;

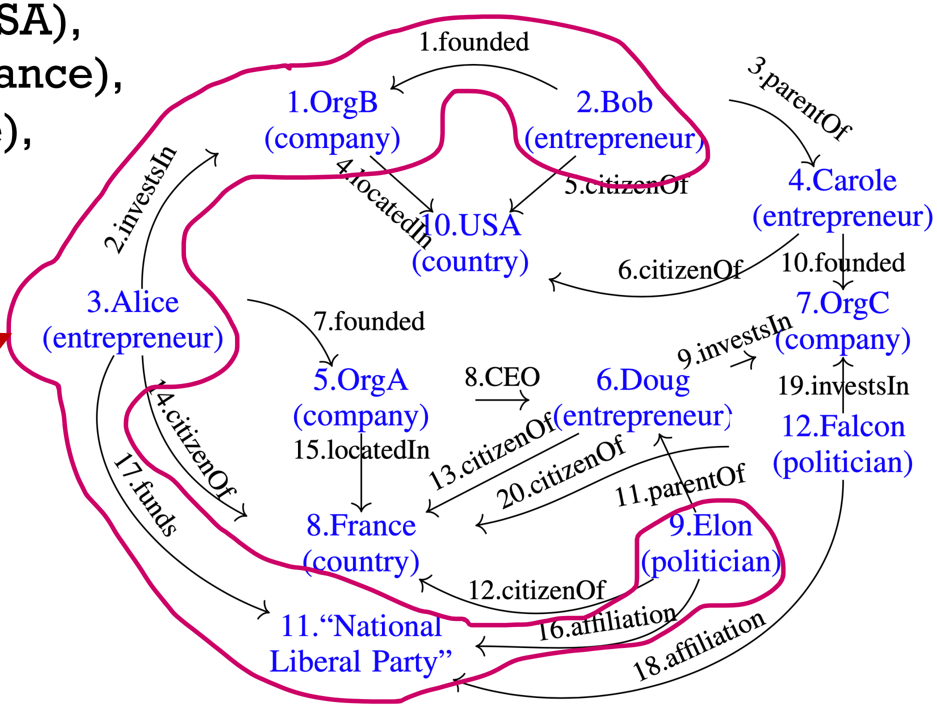
x	y	z
Bob	Alice	Elon
Carole	Doug	Falcon



QUERY SEMANTICS (CTP)

MATCH
(x WHERE x.type = entrepreneur) −[a: citizenOf]→ (b: USA),
(y WHERE y.type= entrepreneur) −[c: citizenOf]→ (d: France),
(z WHERE z.type = politician) −[e: citizenOf]→ (f: France),
(x, y, z, w)
RETURN w;

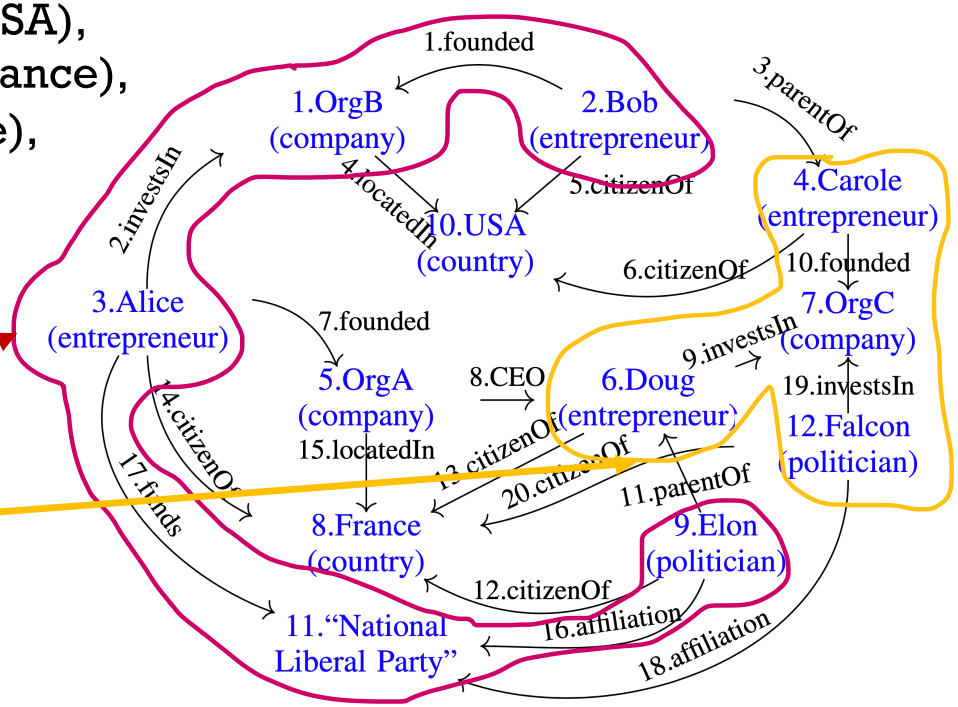
x	y	z	w
Bob	Alice	Elon	
Carole	Doug	Falcon	
Carole	Alice	Falcon	….
and many more			



QUERY SEMANTICS (CTP)

MATCH
(x WHERE x.type = entrepreneur) −[a: citizenOf]→ (b: USA),
(y WHERE y.type= entrepreneur) −[c: citizenOf]→ (d: France),
(z WHERE z.type = politician) −[e: citizenOf]→ (f: France),
(x, y, z, w)
RETURN w;

x	y	z	w
Bob	Alice	Elon	
Carole	Doug	Falcon	
Carole	Alice	Falcon	….
and many more			

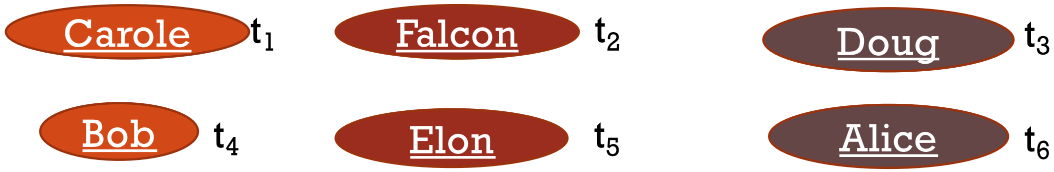


GAM ALGORITHM

Graph integration of structured, semistructured and unstructured data for data journalism, A. Anadiotis et al. Information Systems J. (2022)

Build rooted trees :

- INIT(n)

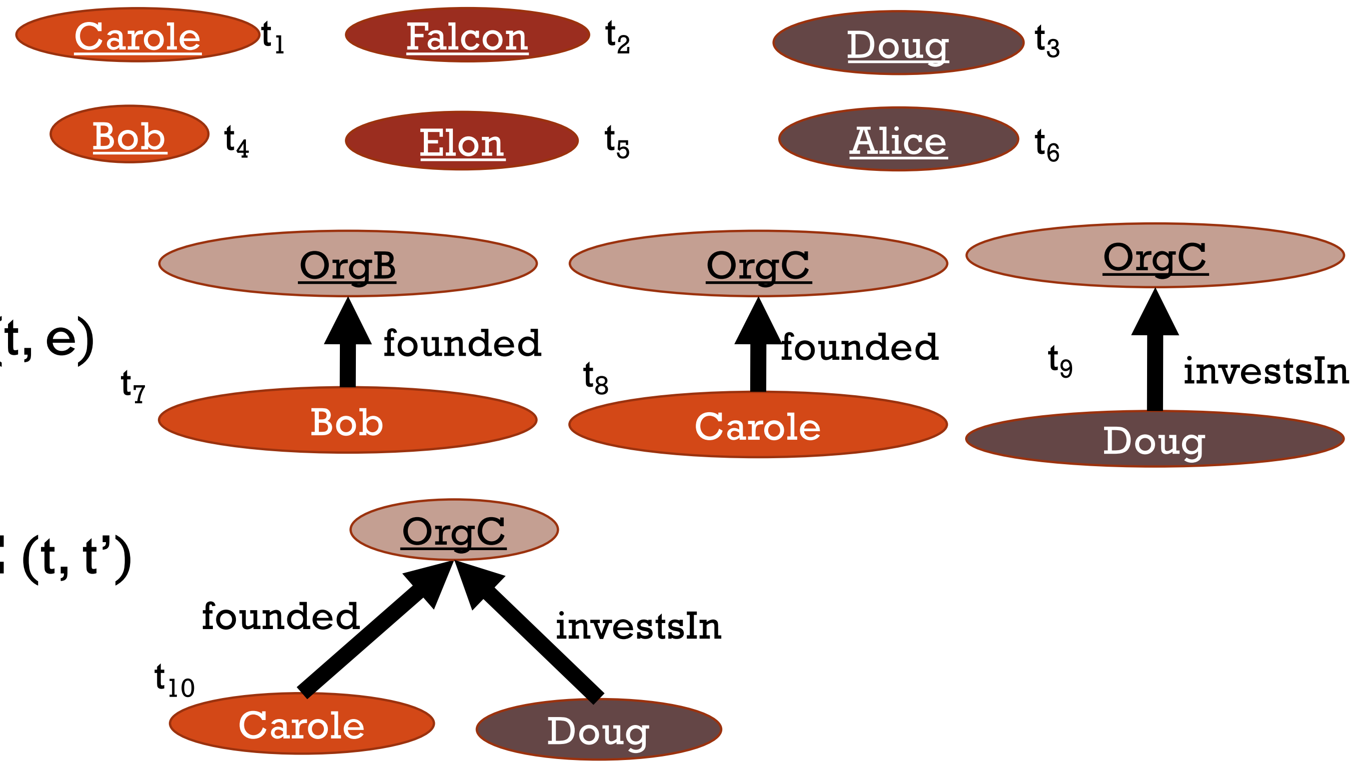


GAM ALGORITHM

Graph integration of structured, semistructured and unstructured data for data journalism, A. Anadiotis et al. Information Systems J. (2022)

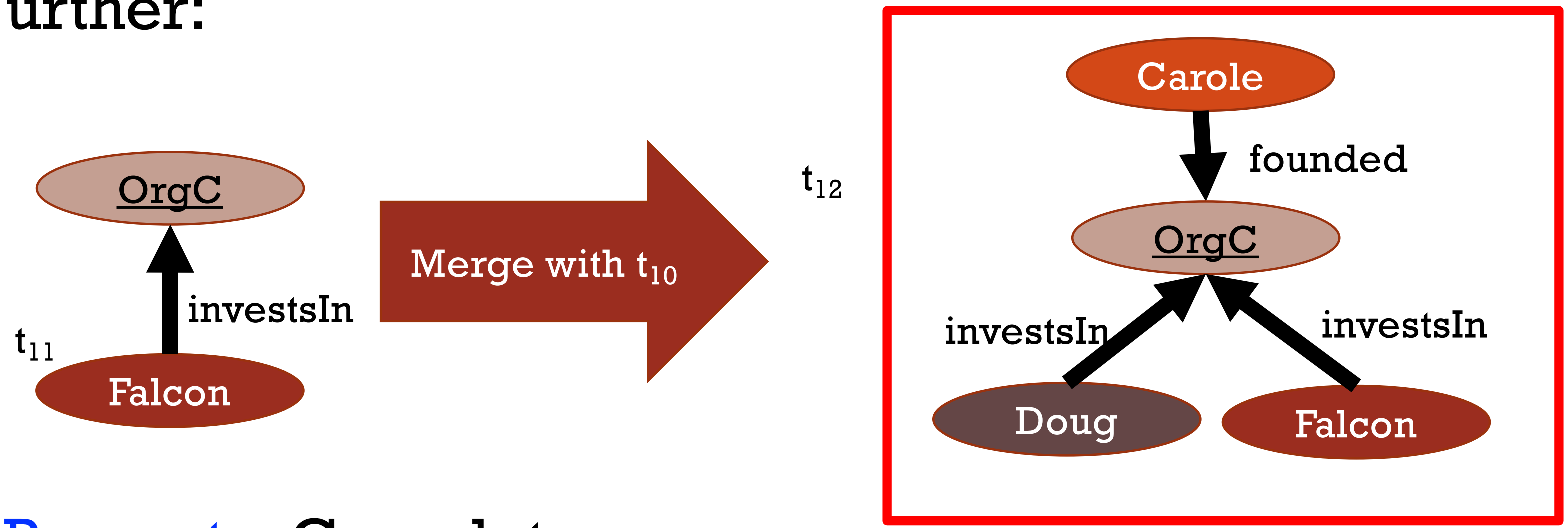
Build rooted trees :

- INIT(n)
- GROW(t, e)
- MERGE (t, t')



GAM ALGORITHM

Further:



- Property: Complete
- Minimal answers *only*.
- Builds all rooted trees. ✗

OPTIMIZATION 1: EDGE SET PRUNING (ESP)

- Keep only the first rooted tree built for the same set of edges.

OPTIMIZATION 1: EDGE SET PRUNING (ESP)

- Keep only the first rooted tree built for the same set of edges.

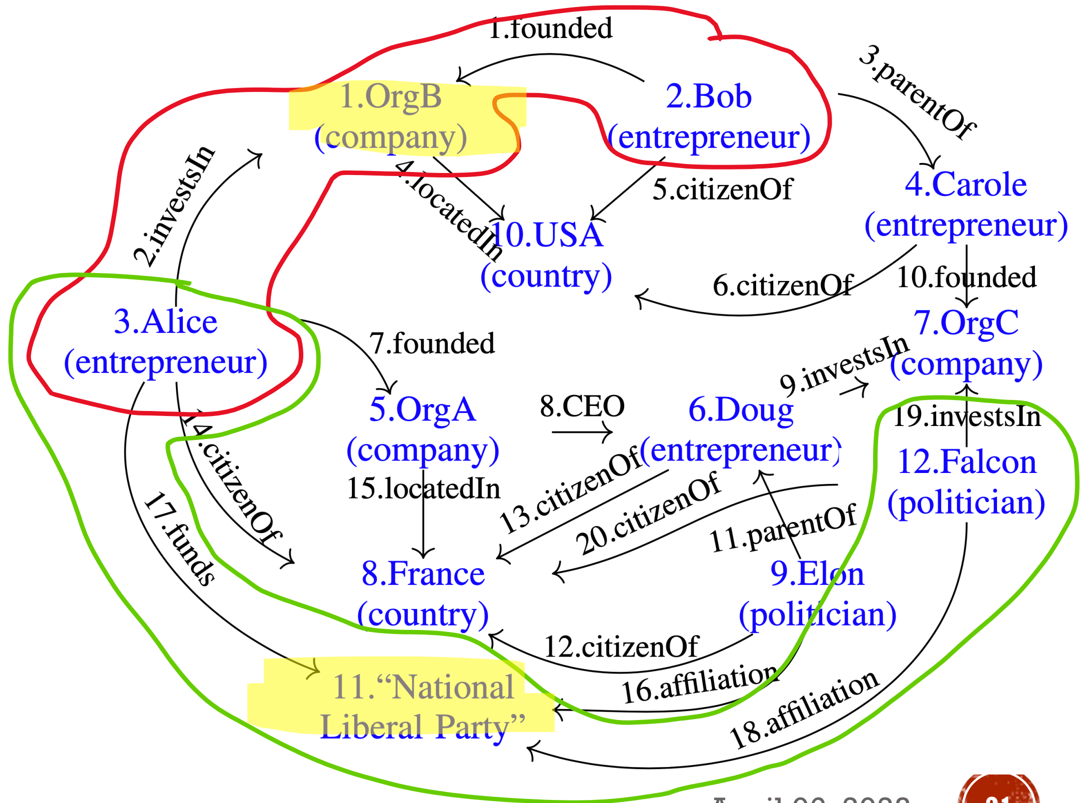
- Property: Complete for 2-input CTPs. ✓

OPTIMIZATION 1: EDGE SET PRUNING (ESP)

- Keep only the first rooted tree built for the same set of edges.

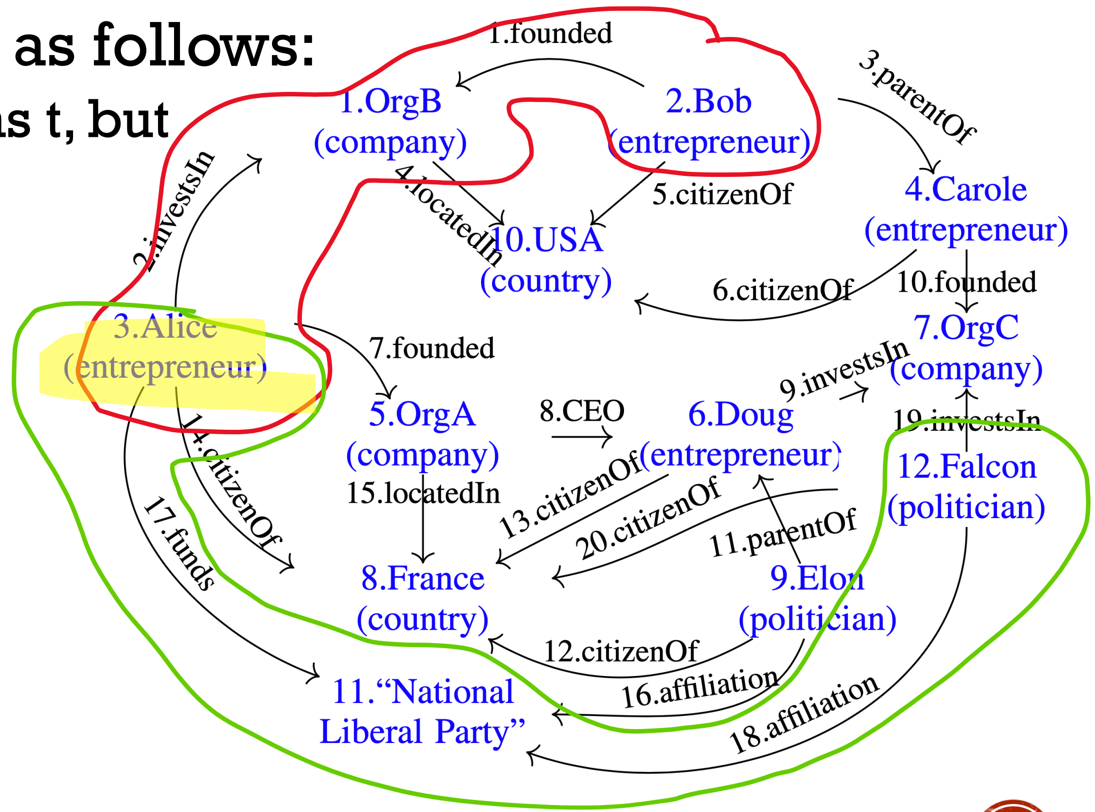
- Property: Complete for 2-input CTPs. ✓

- Incomplete for a CTP of arity >=3. ✗
- Example:
  - Connection between Alice, Bob and Falcon



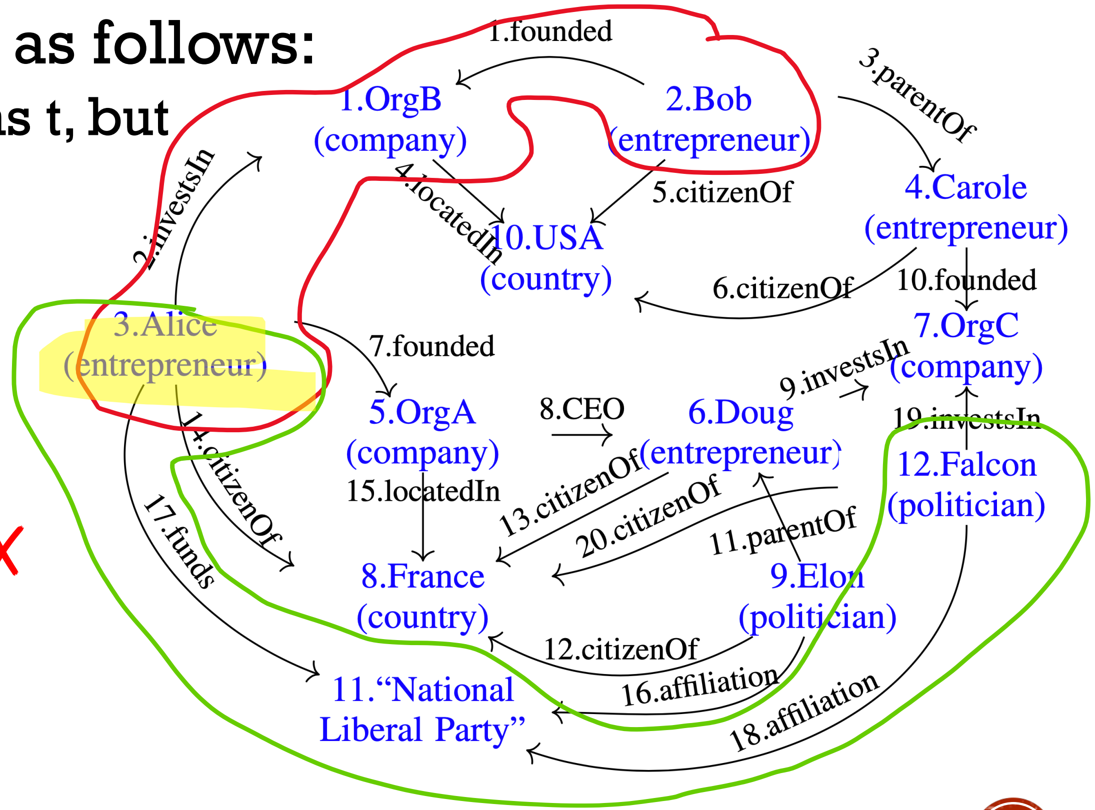
OPTIMIZATION 2: MERGE-ORIENTED ESP (MOESP)

- Build MoESP trees from new tree t as follows:
  - t′ has the same edges (and nodes) as t, but
  - t′ is rooted in a seed r, distinct from the root of t



OPTIMIZATION 2: MERGE-ORIENTED ESP (MOESP)

- Build MoESP trees from new tree t as follows:
  - t′ has the same edges (and nodes) as t, but
  - t′ is rooted in a seed r, distinct from the root of t



- Property: Finds all path results. ✓
- May still fail to find some results. ✗

OPTIMIZATION 3: LIMITED ESP (LESP)

- Signature (s) for each node *n*:
  - #Seed-sets having paths from seeds to *n*.

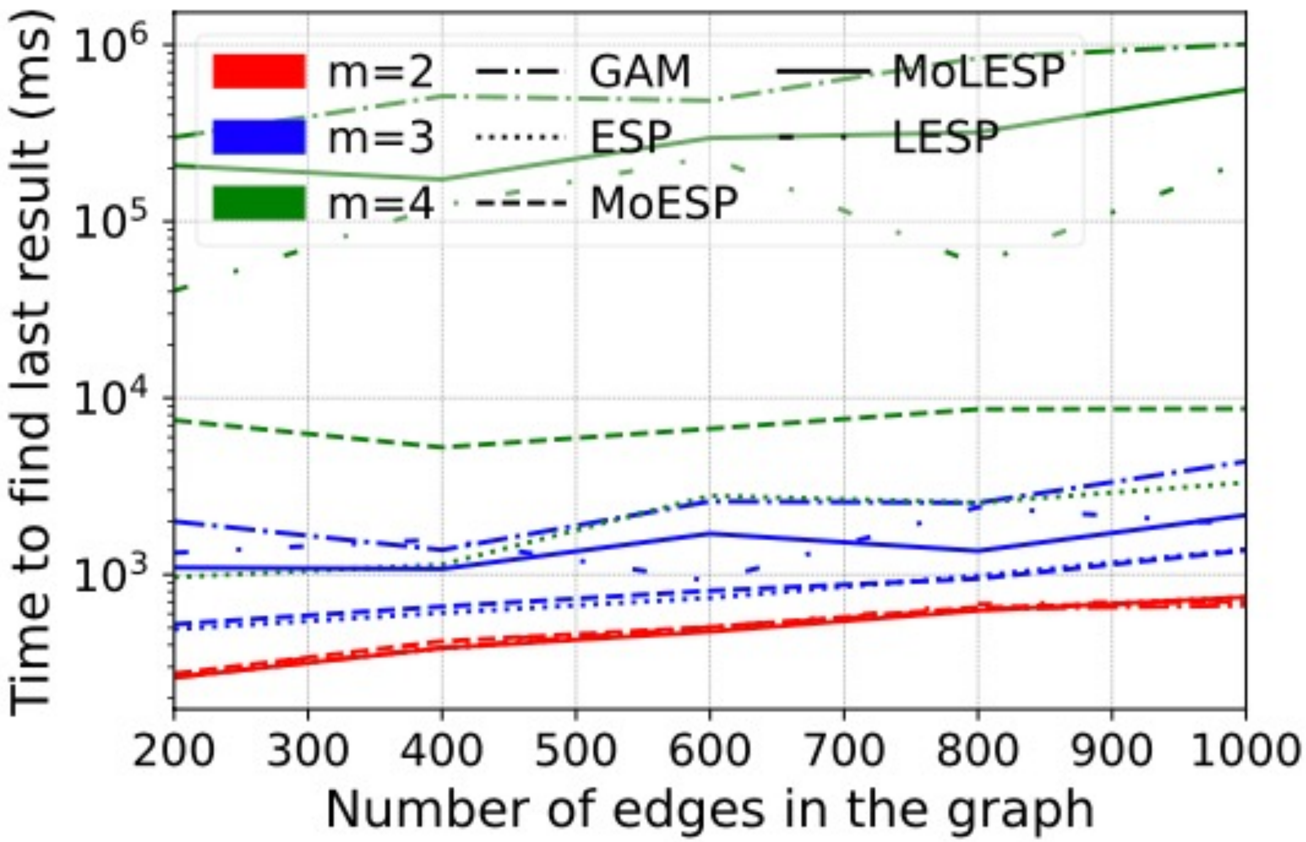
- Limit ESP when the root:
  - Has s>=3.
  - Has 3 or more adjacent edges.

ADDING IT ALL UP - MOLESP

- Variant with ESP, MoESP (inject more trees) and LESP (limit pruning on some Merges).
- Property: Complete for CTPs of arity 3. ✓
- Still incomplete for some CTPs of arity >= 4.
- Property: Identified class of solutions guaranteed to be found for CTPs of any arity (refer to the paper).
  - The frequent cases covered. ✓

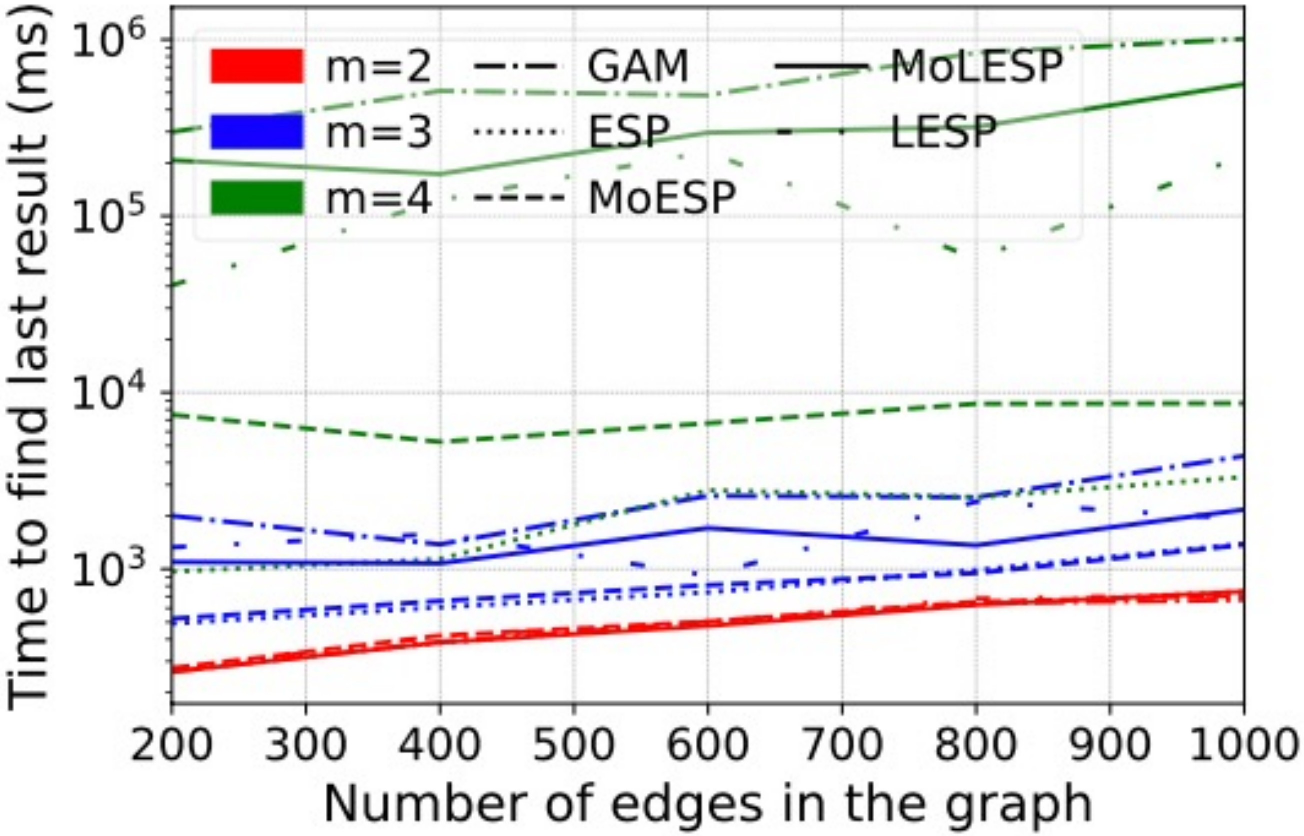
SCALABILITY ON BARABASI ALBERT GRAPHS

Timeout 25 minutes.

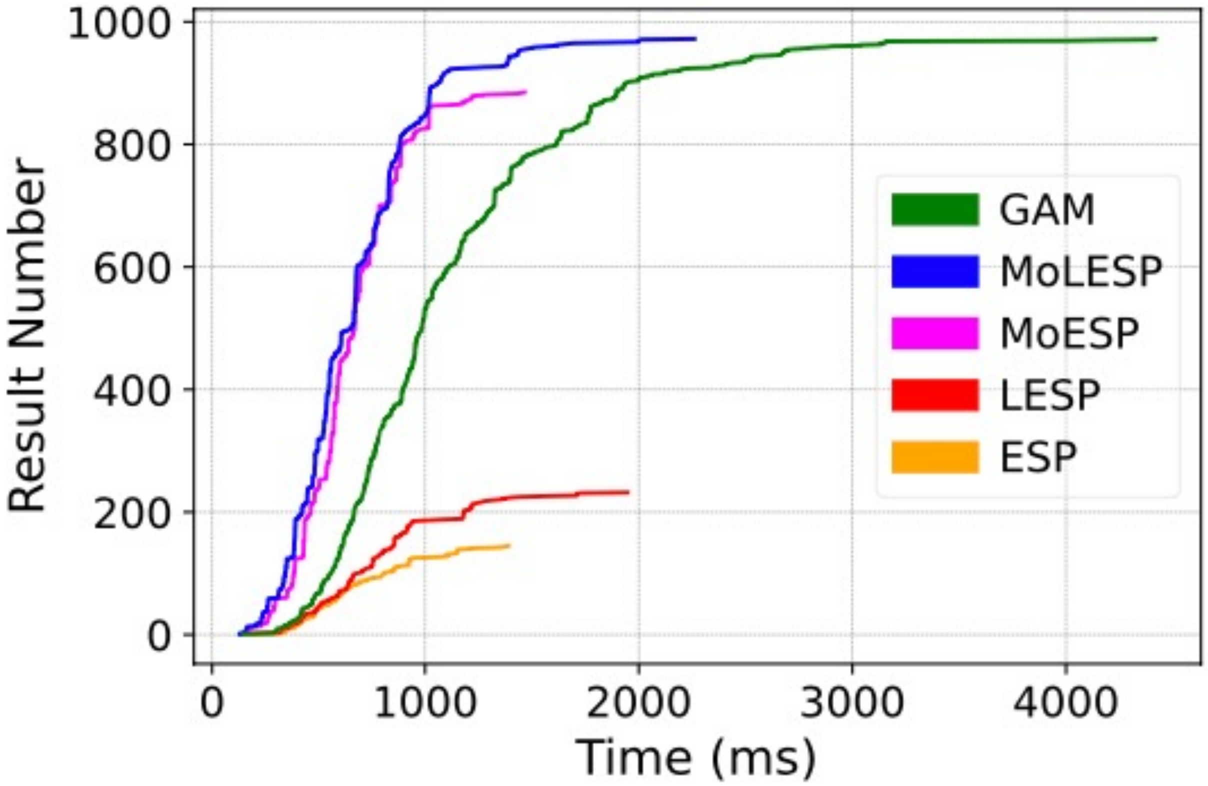


SCALABILITY ON BARABASI ALBERT GRAPHS

MoLESP scales well with the size of Barabasi-Albert graphs.



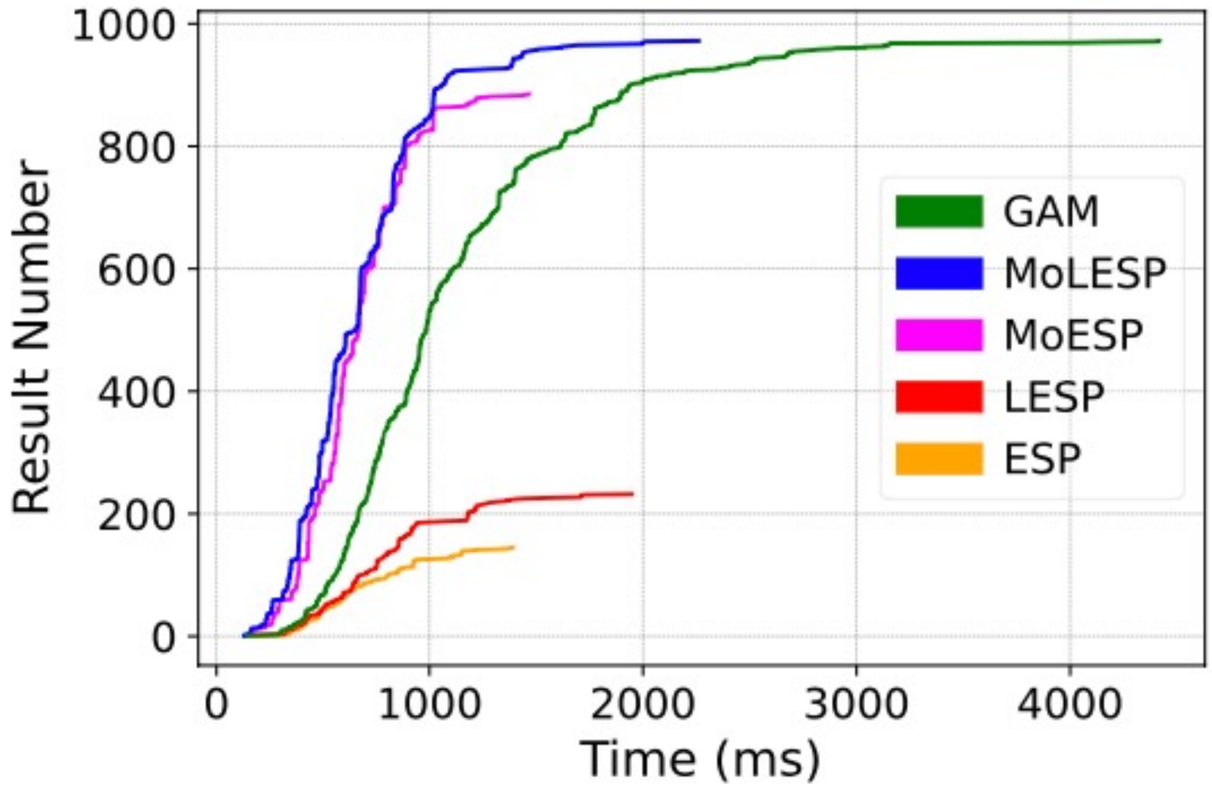
SCALABILITY ON BARABASI ALBERT GRAPHS



m=3

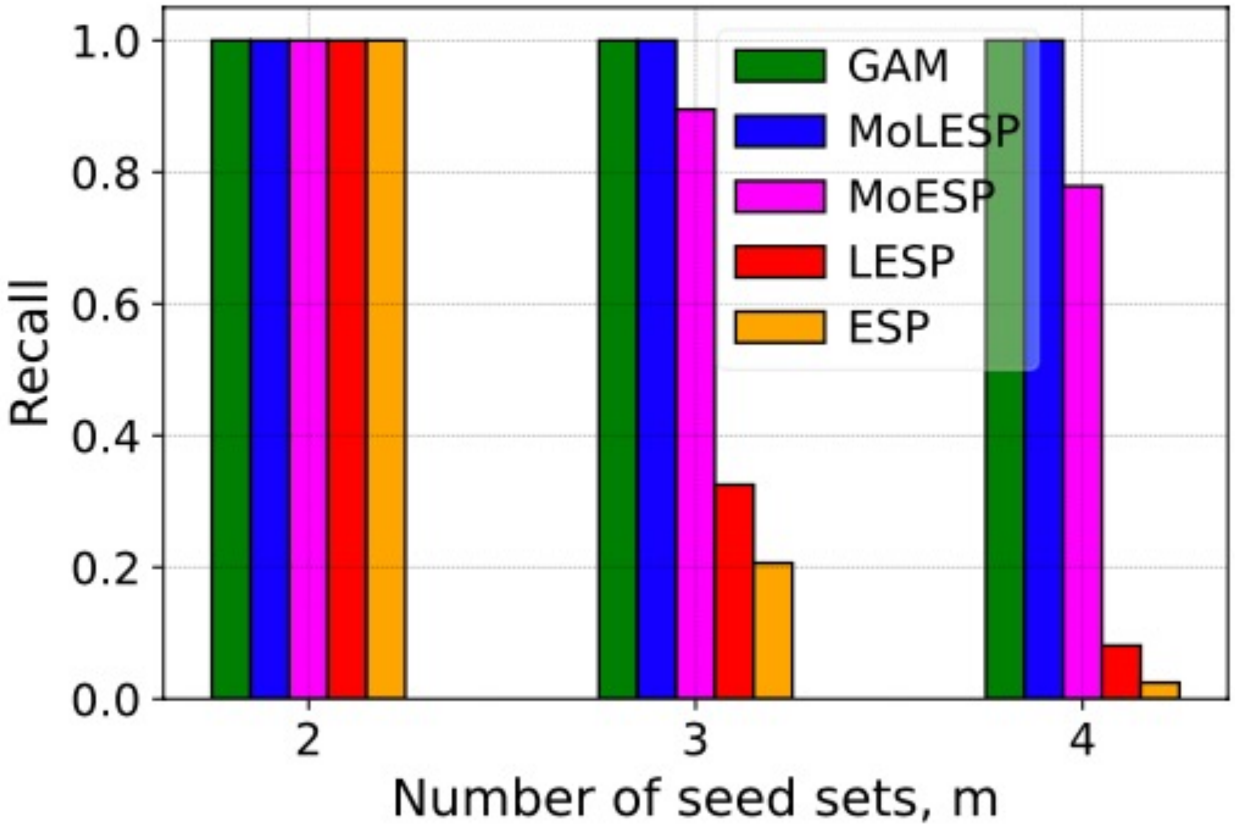
SCALABILITY ON BARABASI ALBERT GRAPHS

MoLESP is 2x faster than GAM, slightly slower than MoESP.



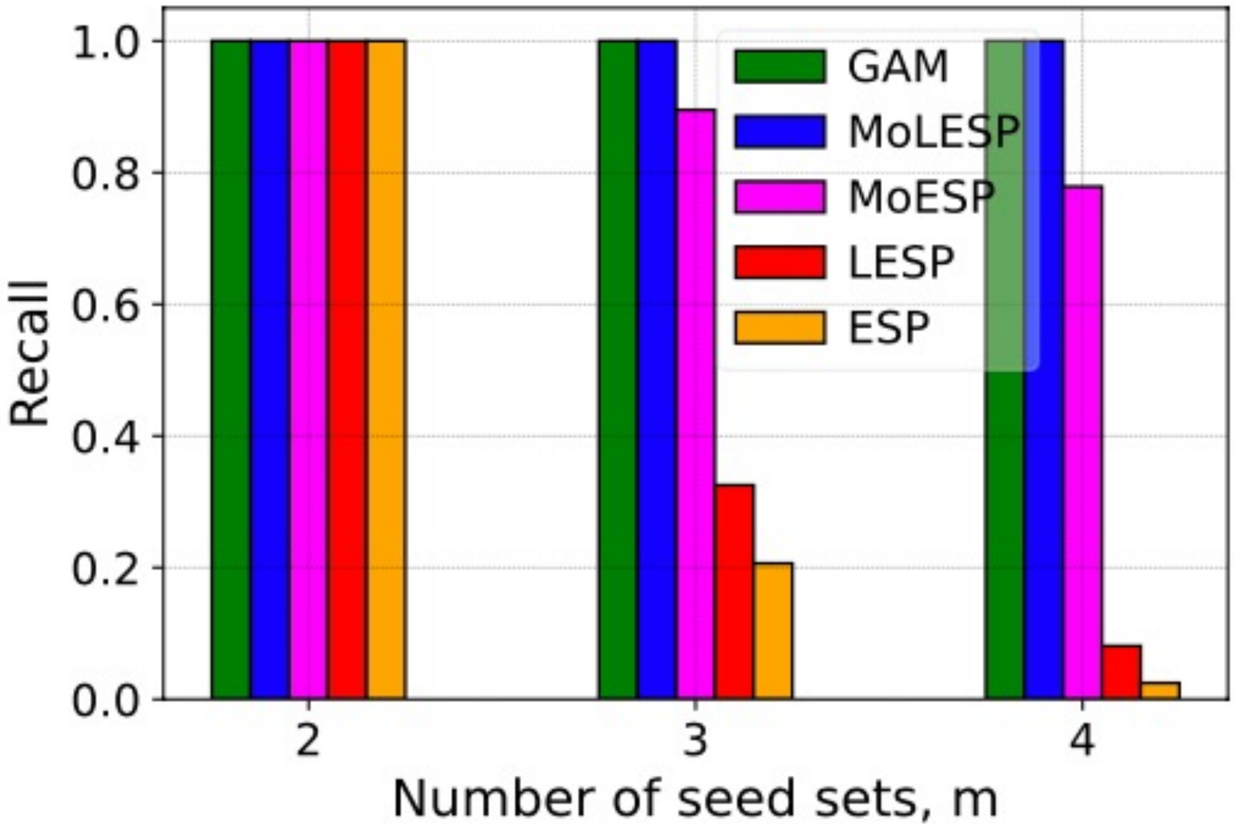
m=3

RECALL ON BARABASI ALBERT GRAPHS



RECALL ON BARABASI ALBERT GRAPHS

MoLESP has a perfect recall even for m=4.

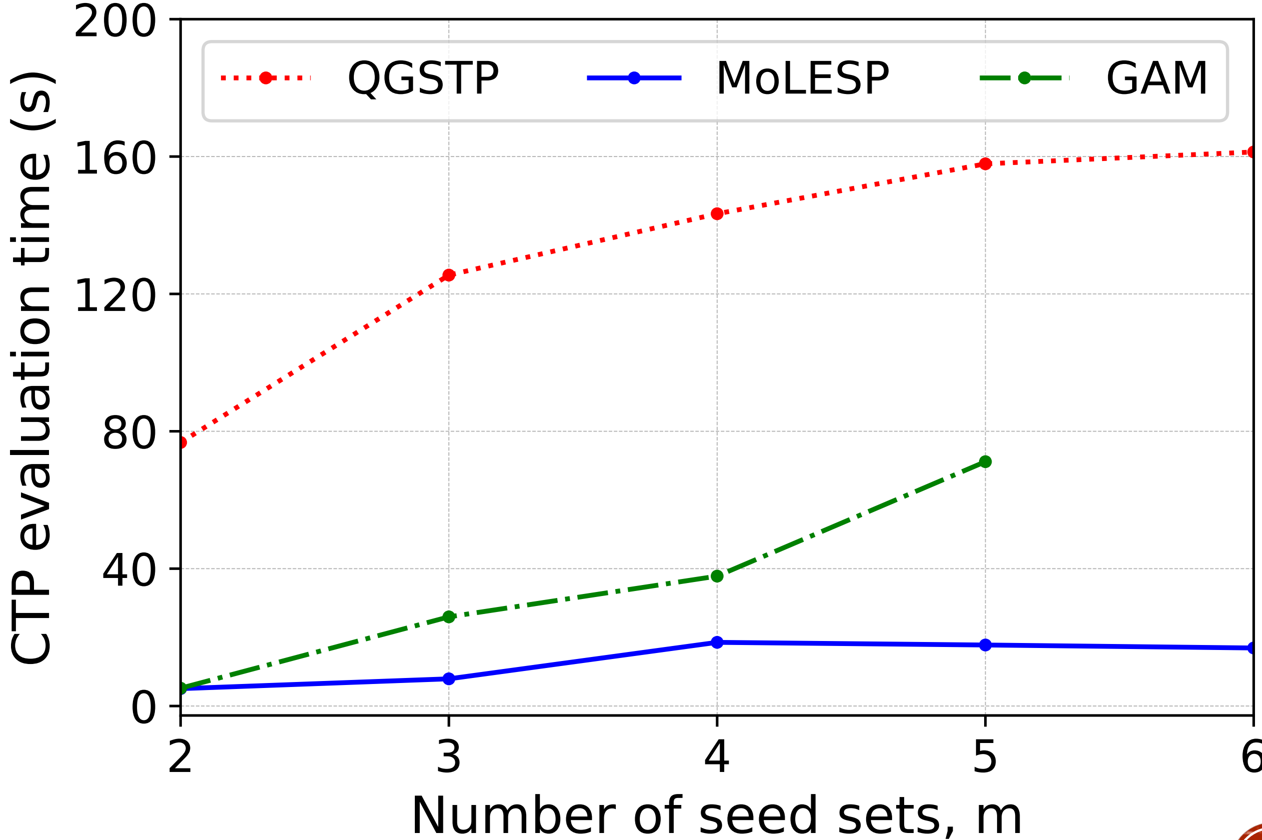


COMPARISON WITH KEYWORD SEARCH SOTA: QGSTP

Kharlamov et al. Efficient Computation of Semantically Cohesive Subgraphs for Keyword-Based Knowledge Graph Exploration. In WWW 2021.

Dataset: DBPedia
#Queries: 312

Timeout 15 minutes

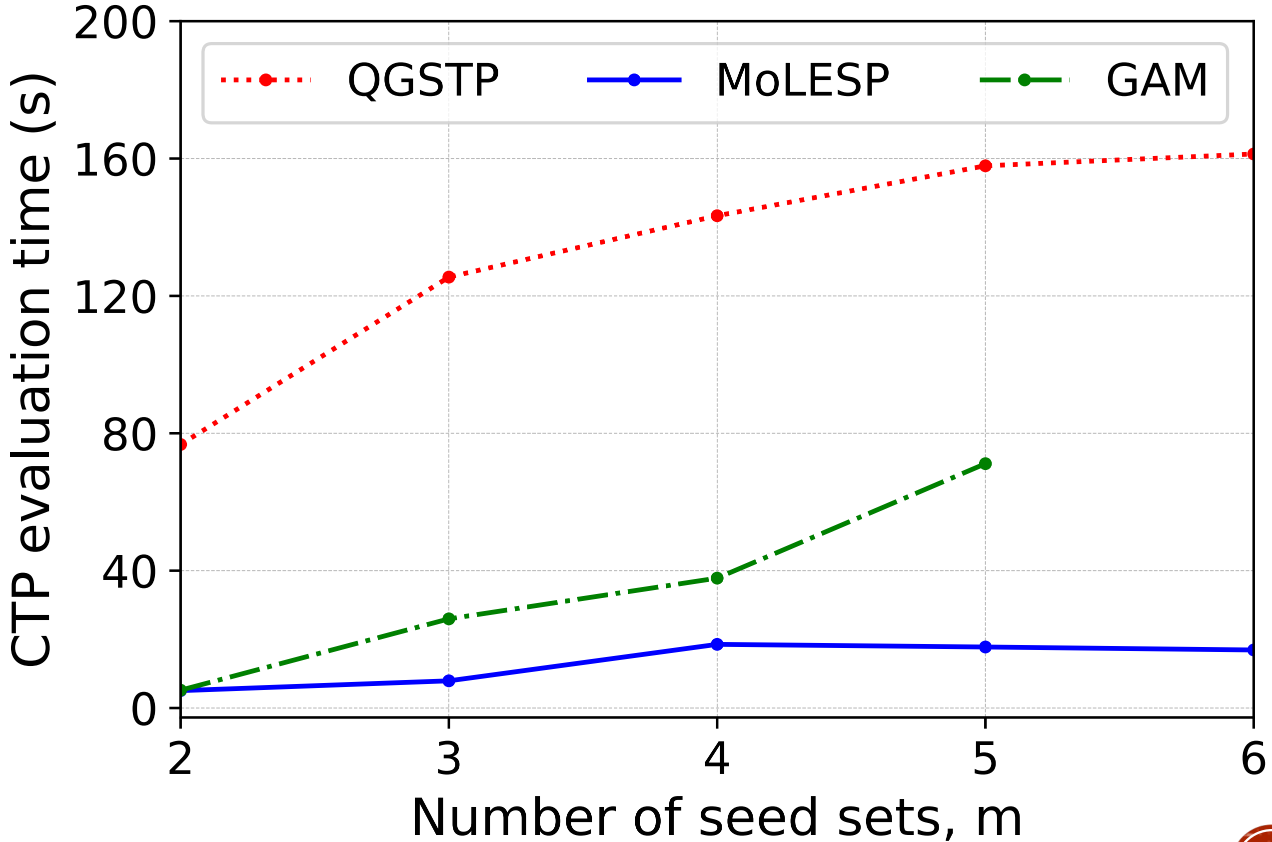


COMPARISON WITH KEYWORD SEARCH SOTA: QGSTP

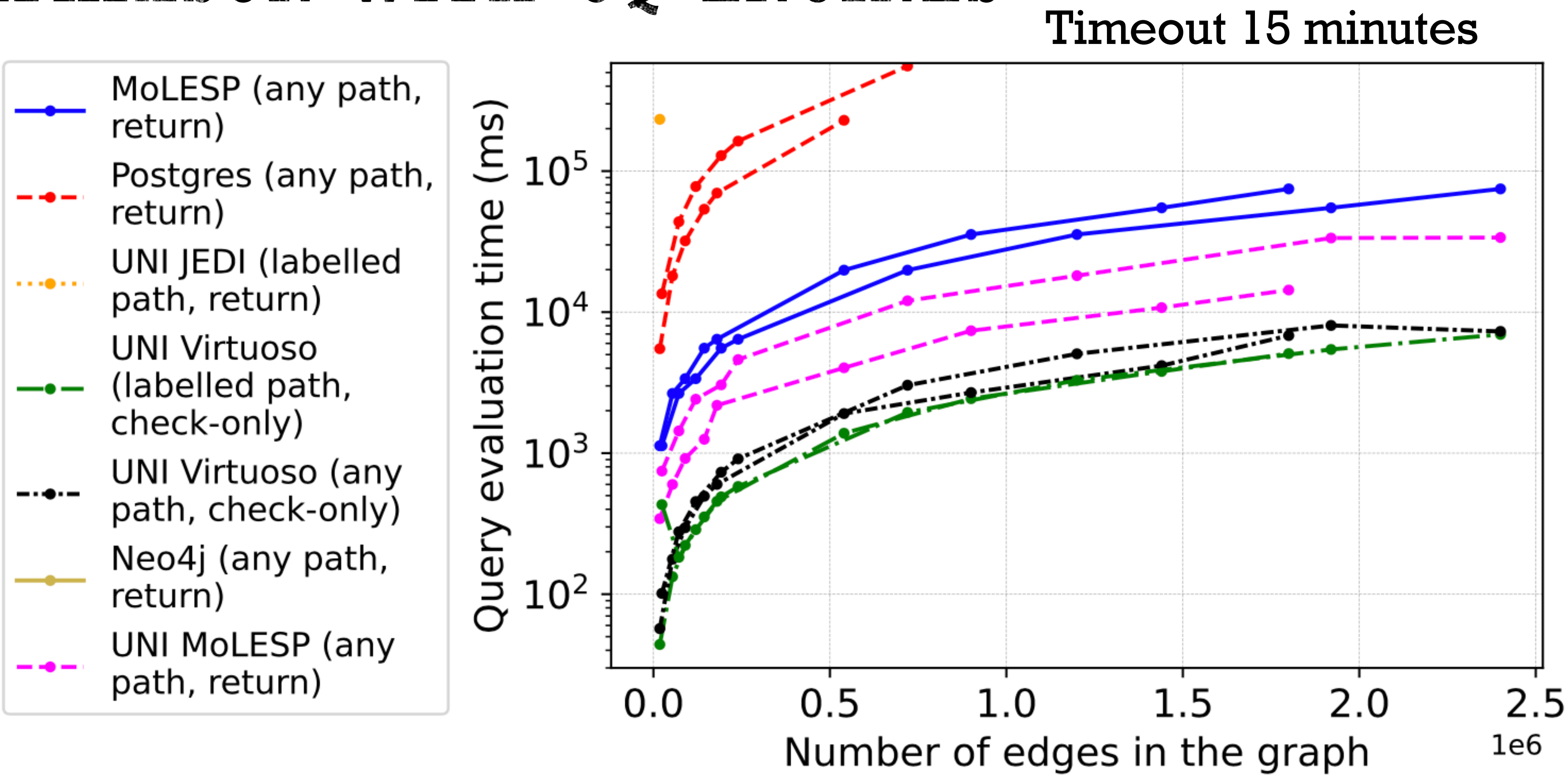
Kharlamov et al. Efficient Computation of Semantically Cohesive Subgraphs for Keyword-Based Knowledge Graph Exploration. In WWW 2021.

Dataset: DBPedia
#Queries: 312

MoLESP is 4x faster in finding result.



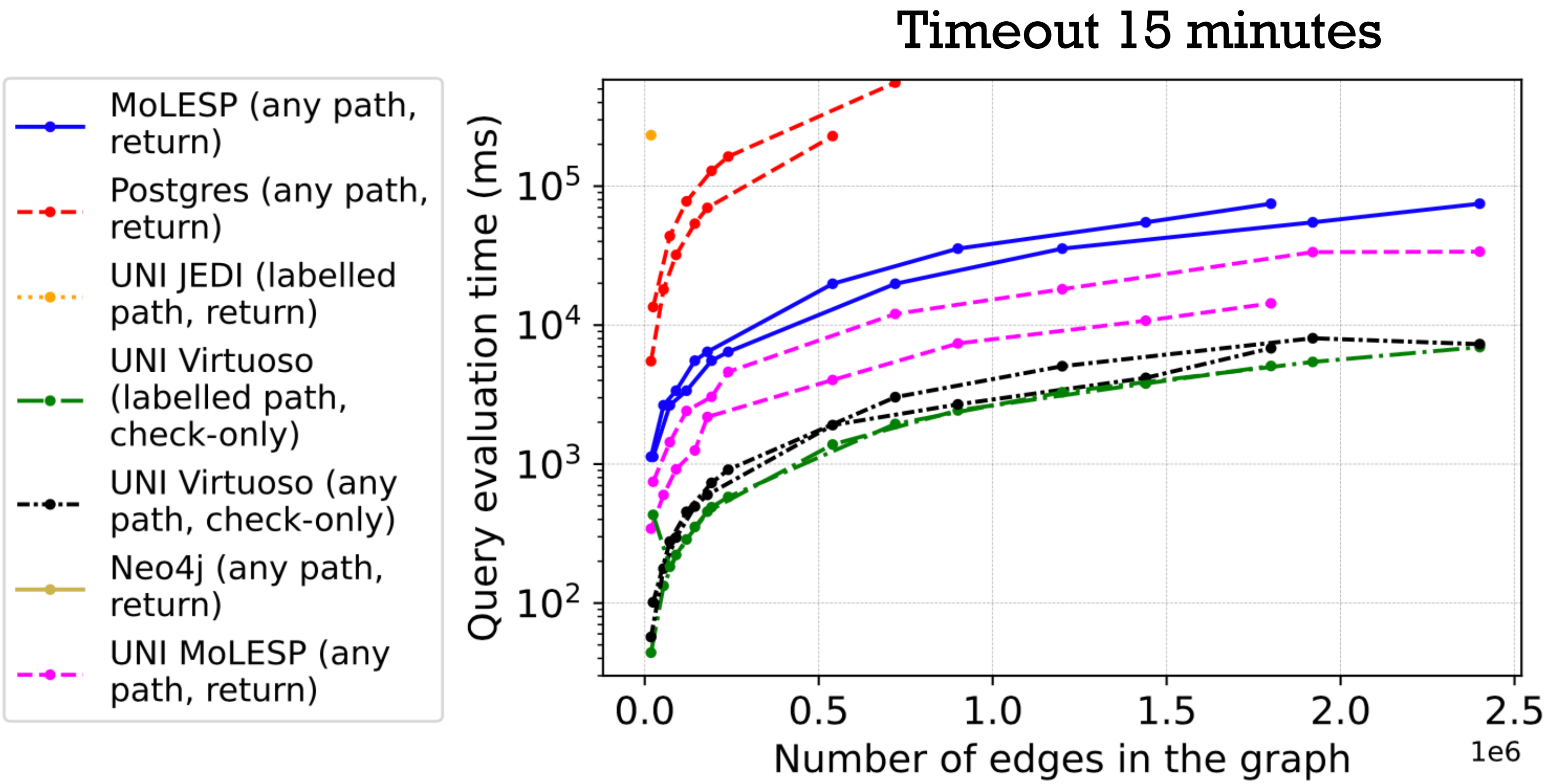
COMPARISON WITH GQ ENGINES



[JEDI] C. Aebeloe, G. Montoya, V. Setty, and K. Hose, "Discovering diversified paths in knowledge bases," *VLDB* 2018.

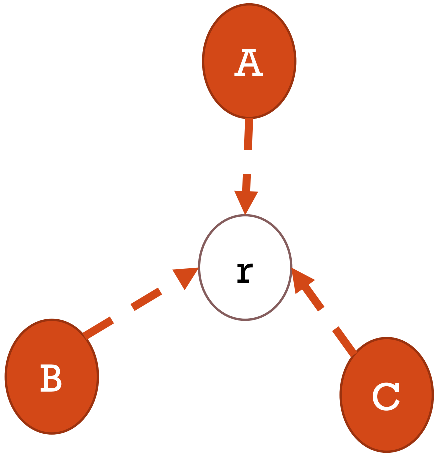
COMPARISON WITH GQ ENGINES

MoLESP is the only feasible path returning algorithm.

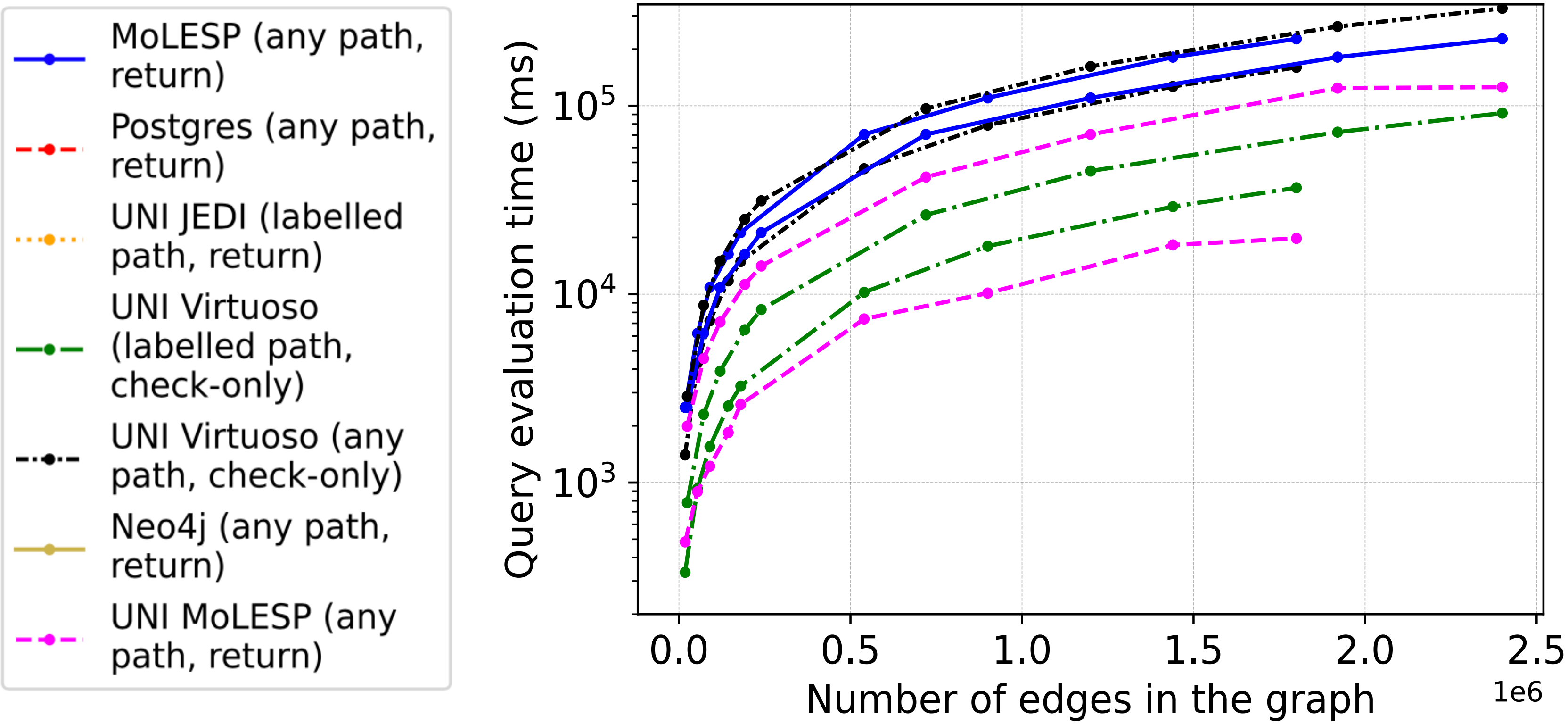


[JEDI] C. Aebeloe, G. Montoya, V. Setty, and K. Hose, “Discovering diversified paths in knowledge bases,” *VLDB* 2018.

COMPARISON WITH GQ ENGINES
PATH STITCHING



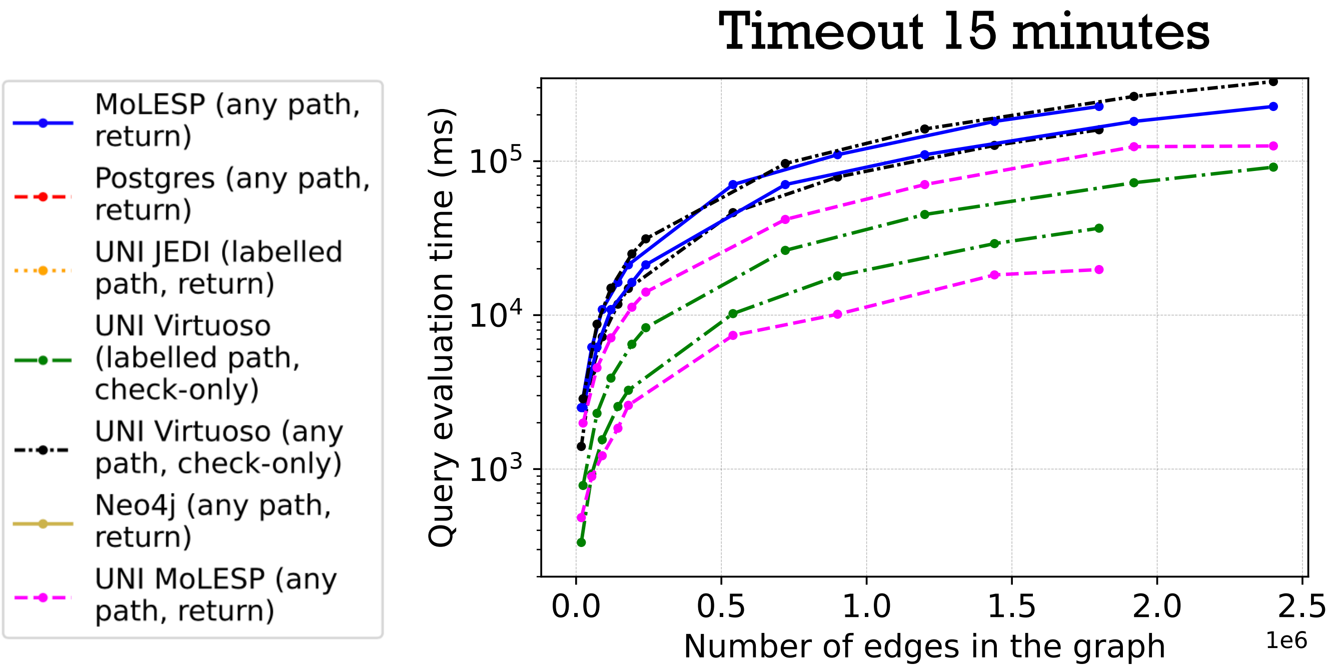
Timeout 15 minutes



[JEDI] C. Aebeloe, G. Montoya, V. Setty, and K. Hose, "Discovering diversified paths in knowledge bases," *VLDB* 2018.

COMPARISON WITH GQ ENGINES PATH STITCHING

MoLESP scales well with the graph size and the cost of adding CTP is minimal.



[JEDI] C. Aebeloe, G. Montoya, V. Setty, and K. Hose, "Discovering diversified paths in knowledge bases," *VLDB* 2018.

CONCLUSION

- Extension to GPML by using CTPs.
  - Supports asking for connecting trees.
- MoLESP: Efficient search algorithm for the connecting trees.
- Future work:
  - Smart execution strategies for jointly optimizing GPs and CTPs.
  - Optimized execution of multiple CTPs.

THANK YOU

Project Web site: https://team.inria.fr/cedar/connectionlens/

Code and datasets for this paper: https://gitlab.inria.fr/cedar/extended-graph-querying