SEATTLE WA, USA SIGMOD PODS 2023

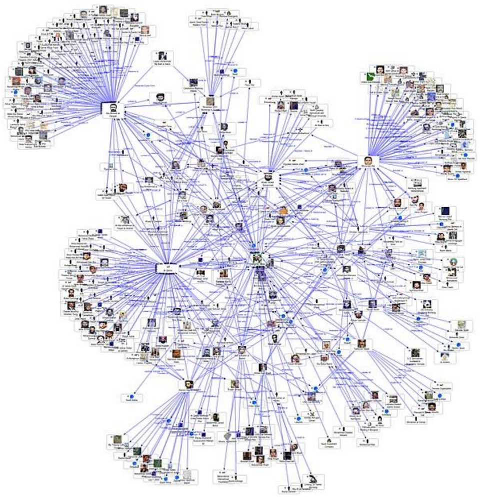
An Overview of Reachability Indexes on Graphs

Chao Zhang[1], Angela Bonifati[2], and M. Tamer Özsu[1]

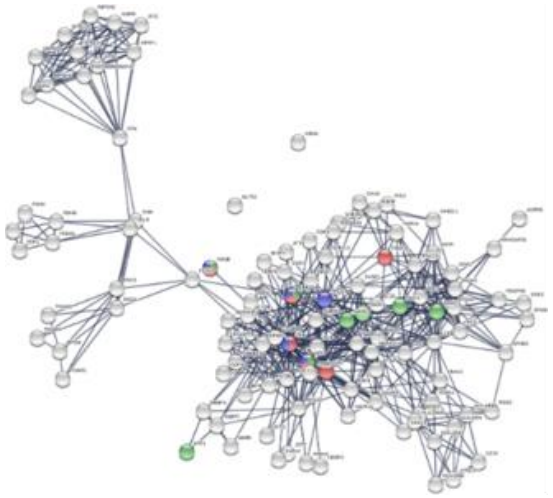
[1]University of Waterloo

[2]Lyon 1 University

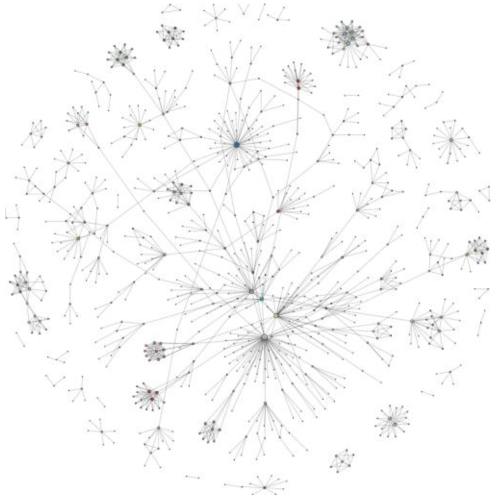
UNIVERSITY OF WATERLOO | DSg Data Systems Group | UCBL Lyon 1



Social networks



Biological networks



Web graphs …

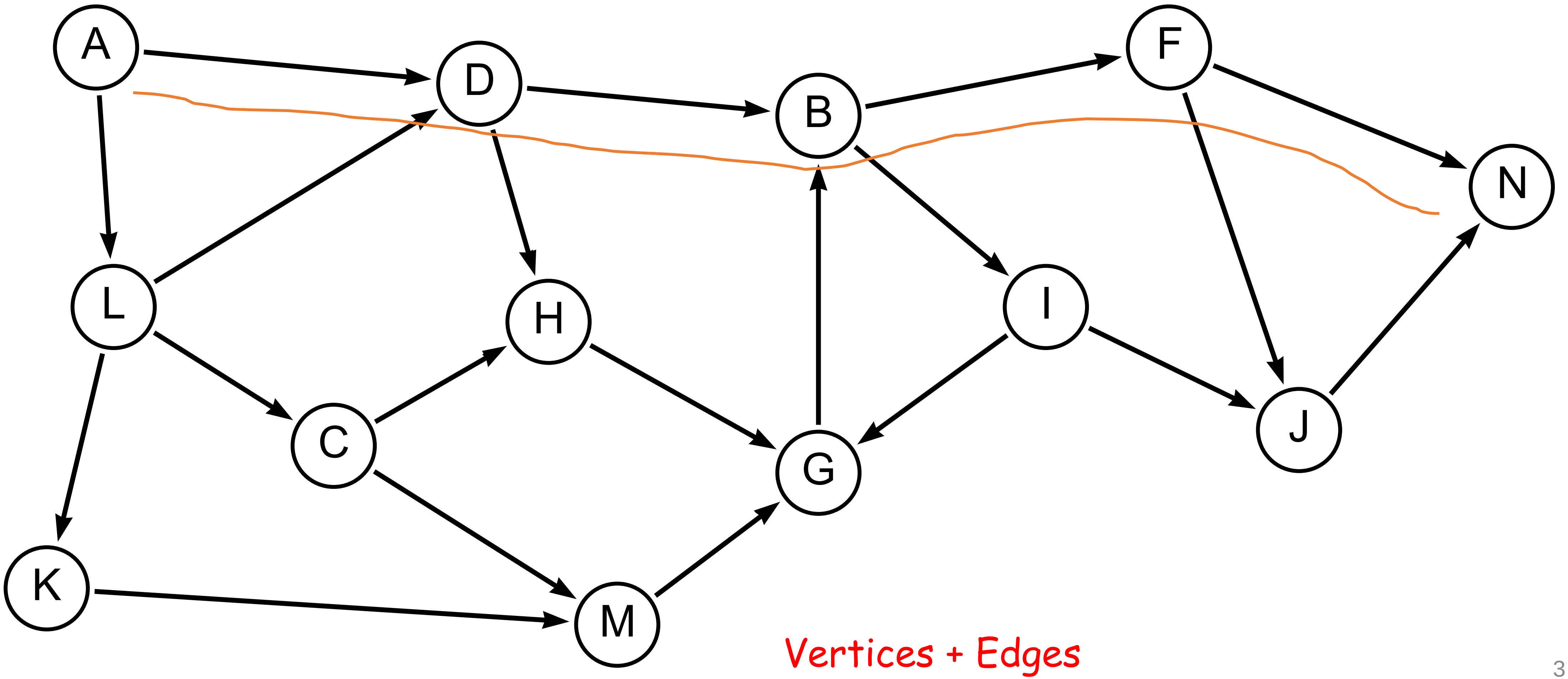
Graphs are everywhere

One of the fundamental graph processing operations [Sah20]: **Reachability Queries**

[Sah20] S. Sahu et al. The ubiquity of large graphs and surprising challenges of graph processing: extended survey. VLDB J. 29(2-3): 595-618 (2020)

Plain Graphs

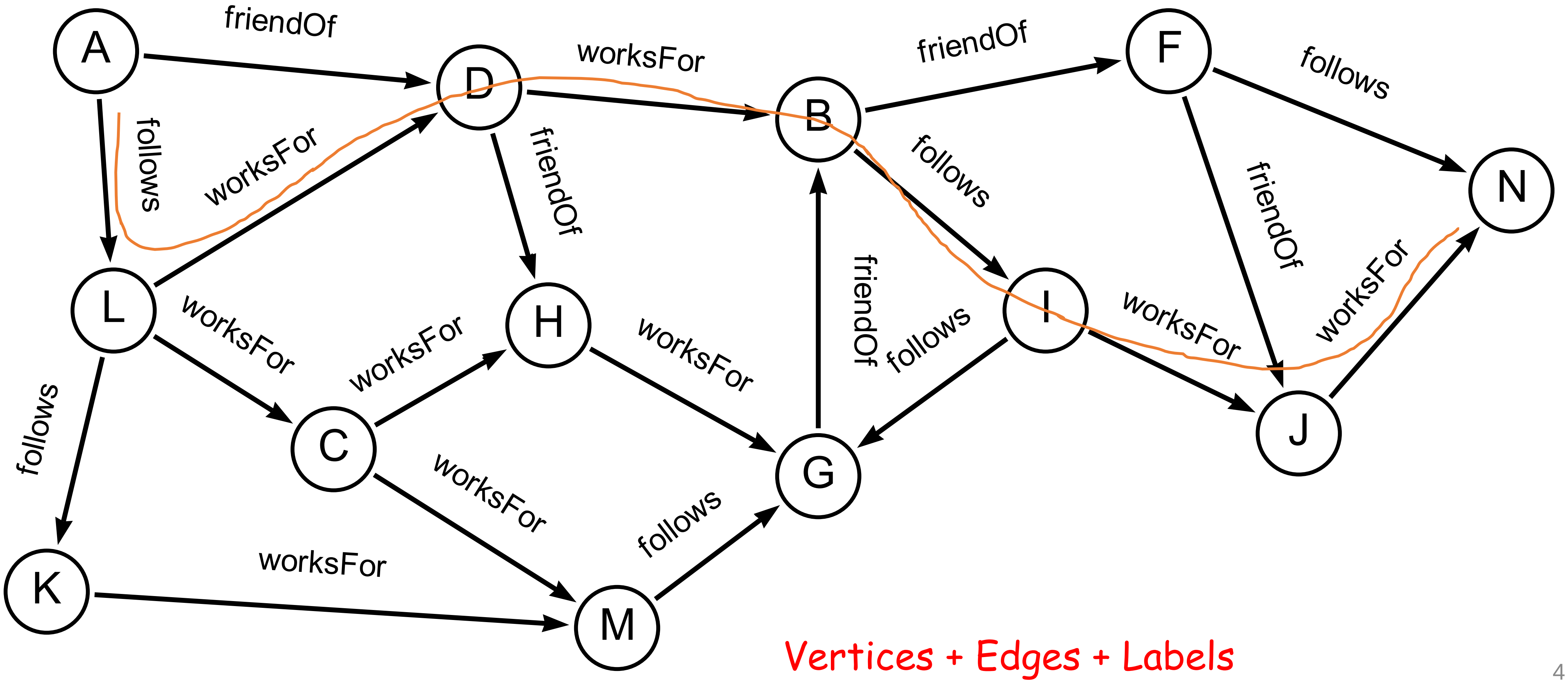
Reachability Query: $Q_r(A, N)$ = True



Vertices + Edges

Edge-Labeled Graphs

Reachability Query: $Q_r(A, N, (follows \cup worksFor)^*)$ = True

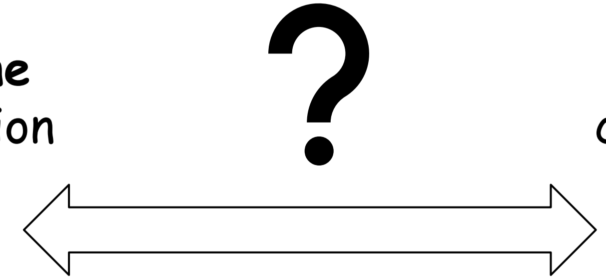


Vertices + Edges + Labels

Reachability Indexes

Full **online** computation

BFS



Full **offline** computation

Transitive Closure

Striking the balance between transitive closure and online traversal

- Two types of reachability indexes
  - Plain graphs: plain reachability indexes
  - Edge-labeled graphs: path-constrained reachability indexes

Outline

1. Plain Reachability Indexes
   a) Tree-Cover Indexes
   b) 2-Hop Indexes
   c) Approximated Transitive Closures
2. Path-Constrained Reachability Indexes
   a) Indexes for Alternation-Based Queries
   b) Indexes for Concatenation-Based Queries
3. Open Challenges

Plain Reachability Indexes

Indexing technique	Framework	Index type	Input	Dynamic	References
Tree cover	Tree cover	Complete	DAG	No	[Agr89]
Tree+SSPI	Tree cover	Partial	DAG	No	[Che05]
Dual labeling	Tree cover	Complete	DAG	No	[Wan06]
GRIPP	Tree cover	Partial	General Graph	No	[Tri07]
Path-Tree	Tree cover	Complete	DAG	Yes	[Jin08,Jin11]
GRAIL	Tree cover	Partial	DAG	No	[Yil10]
Ferrari	Tree cover	Partial	DAG	No	[Seu13]
DAGGER	Tree cover	Partial	DAG	Yes	[Yil13]
2-Hop	2-Hop	Complete	General Graph	No	[Coh02]
Ralf et al.	2-Hop	Complete	General Graph	Yes	[Sch05]
3-Hop	2-Hop	Complete	DAG	No	[Jin09]
U2-Hop	2-Hop	Complete	DAG	Yes	[Bra10]
Path-Hop	2-Hop	Complete	DAG	No	[Cai10]
TFL	2-Hop	Complete	DAG	No	[Che13]
DL	2-Hop	Complete	General Graph	No	[Jin13]
PLL	2-Hop	Complete	General Graph	No	[Yan13]
TOL	2-Hop	Complete	DAG	Yes	[Zhu14]
DBL	2-Hop	Partial	General Graph	Yes	[Lyu21]
O'Reach	2-Hop	Partial	DAG	No	[Han21]
IP	Approximated TC	Partial	DAG	Yes	[Wei14,Wei18]
BFL	Approximated TC	Partial	DAG	No	[Su17]
HL	-	Complete	DAG	No	[Jin13]
Feline	-	Partial	DAG	No	[Vel14]
Preach	-	Partial	DAG	No	[Mer14]

Complete index: index-only query processing

Partial index: index-graph query processing

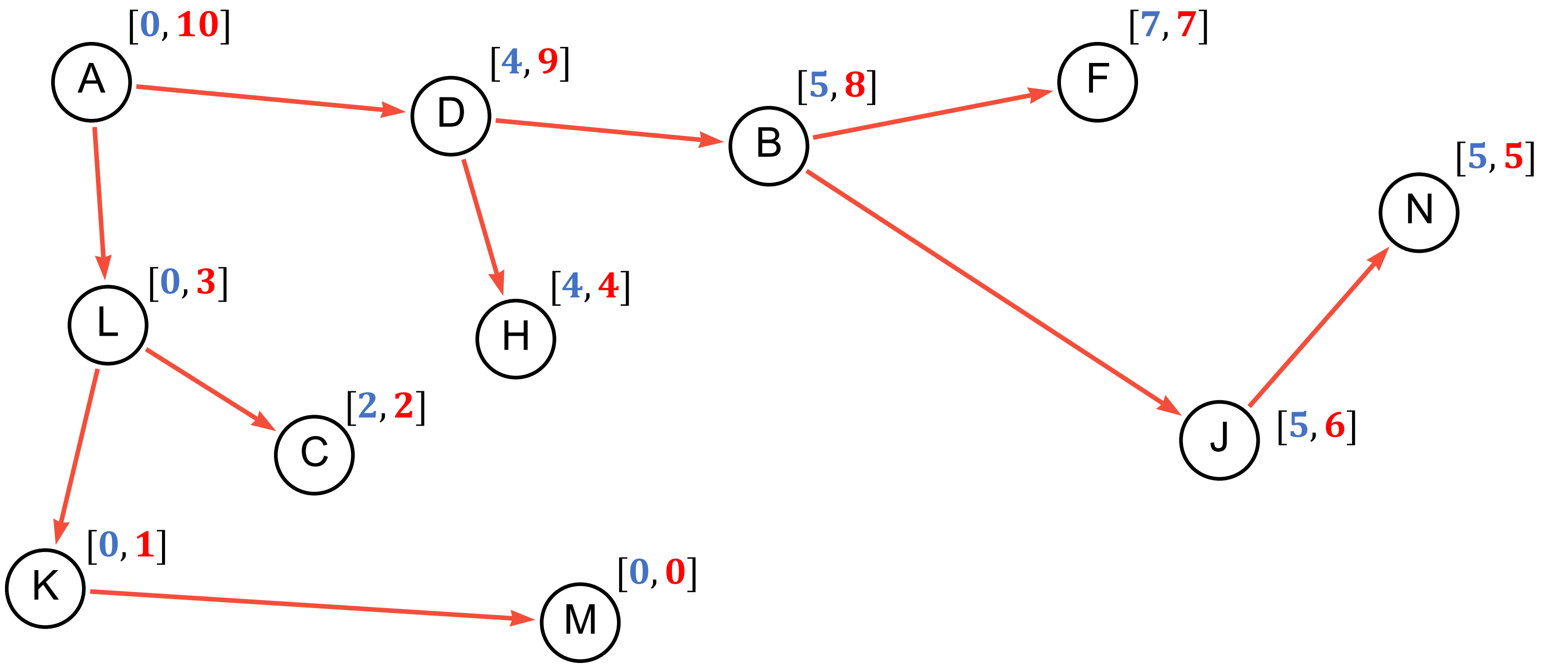
Three index frameworks:

- Tree cover
- 2-Hop
- Approximated TC

Outline

1. Plain Reachability Indexes
    a) Tree-Cover Indexes
    b) 2-Hop Indexes
    c) Approximated Transitive Closures
2. Path-Constrained Reachability Indexes
    a) Indexes for Alternation-Based Queries
    b) Indexes for Concatenation-Based Queries
3. Open Challenges

Interval Labeling



Assign an interval $[a_v, b_v]$ to each vertex v, denoted as $\mathcal{L}_v$

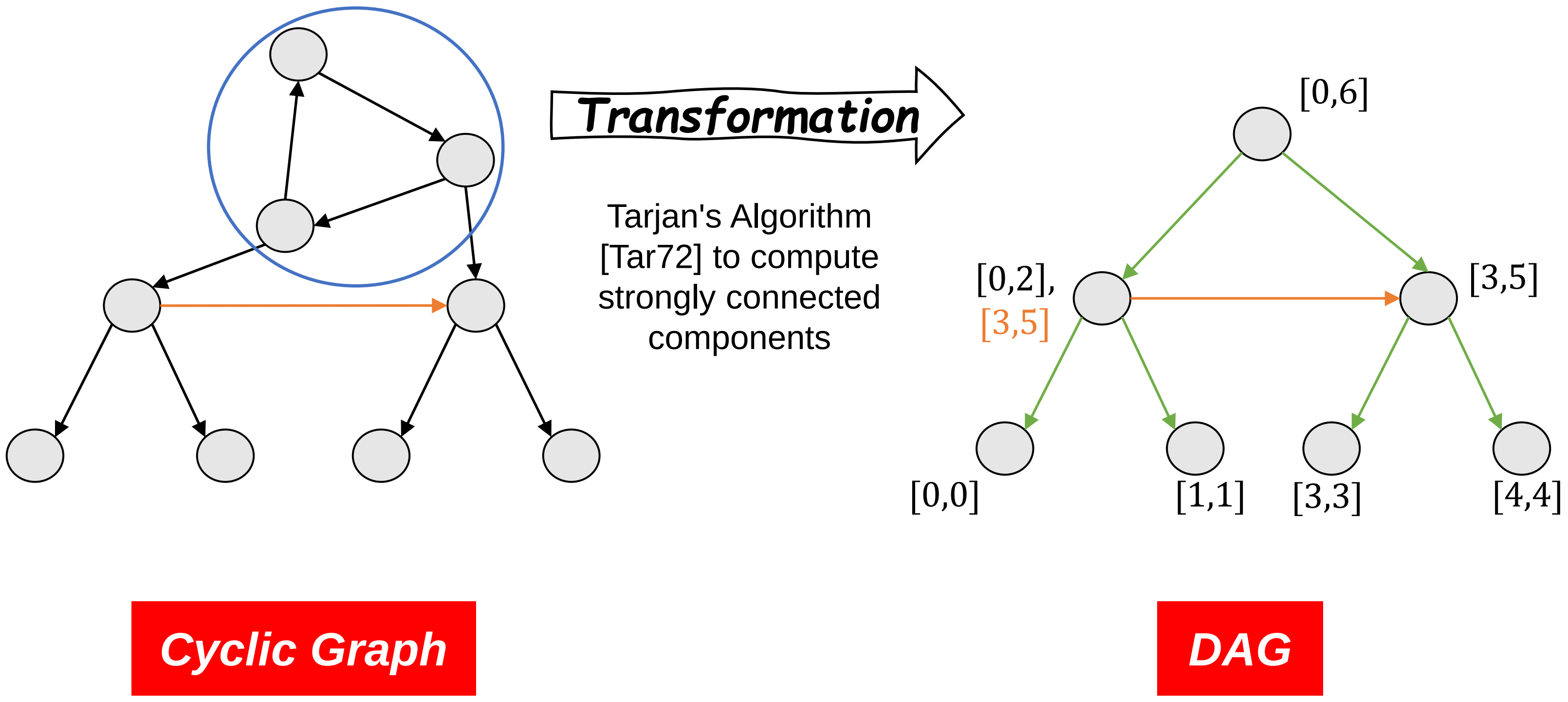
a_v: The lowest postorder number of all the descendants of v

b_v: Postorder number of v

$Q_r(A, J)$: True

- $b_J \in \mathcal{L}_A$, i.e., $[0,10]$ contains 6

Tree Cover Index



Reachability in DAG:

- Interval labeling for the spanning trees in a DAG
- Inheriting intervals due to non-tree edges

[Tar72] R. Tarjan. Depth-First Search and Linear Graph Algorithms. SIAM J. Comput. 1(2): 146-160 (1972)
[Agr89] R. Agrawal et al. Efficient Management of Transitive Relationships in Large Data and Knowledge Bases. SIGMOD Conference 1989: 253-262

Reducing the Number of Intervals

- Bottleneck of Tree-Cover index:
  - A large number of intervals due to non-tree edges
- Bounding the number of intervals
  - GRAIL [Yil10], and Ferrari [Seu13]
  - Partial indexes
  - Querying processing:
    - online search accelerated by leveraging the partial indexes

[Yil10] H. Yildirim et al. GRAIL: Scalable Reachability Index for Large Graphs. Proc. VLDB Endow. 3(1): 276-284 (2010)
[Seu13] S. Seufert et al. FERRARI: Flexible and efficient reachability range assignment for graph indexing. ICDE 2013: 1009-1020

Outline

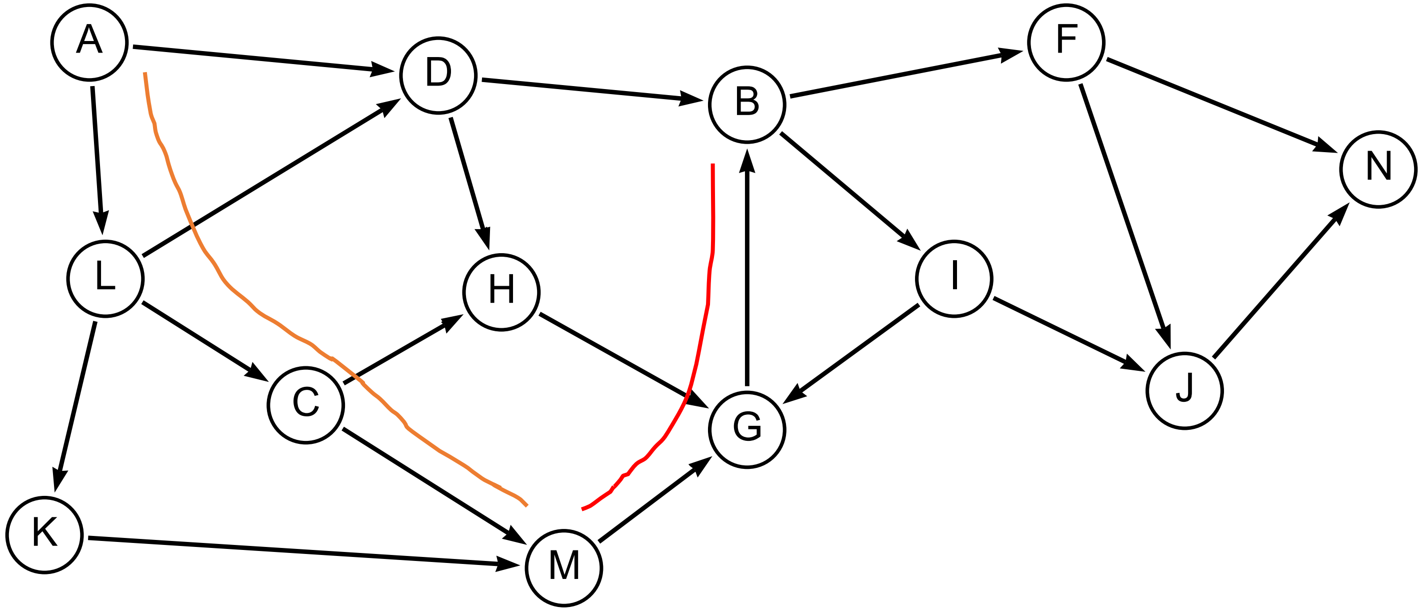
1. Plain Reachability Indexes
    a) Tree-Cover Indexes
    b) 2-Hop Indexes
    c) Approximated Transitive Closures
2. Path-Constrained Reachability Indexes
    a) Indexes for Alternation-Based Queries
    b) Indexes for Concatenation-Based Queries
3. Open Challenges

2-Hop Labeling

Assigning $L(v) = (L_{in}(v), L_{out}(v))$ for each v, such that

$\forall u \in L_{in}(v)$, $\exists$ a path from u to v

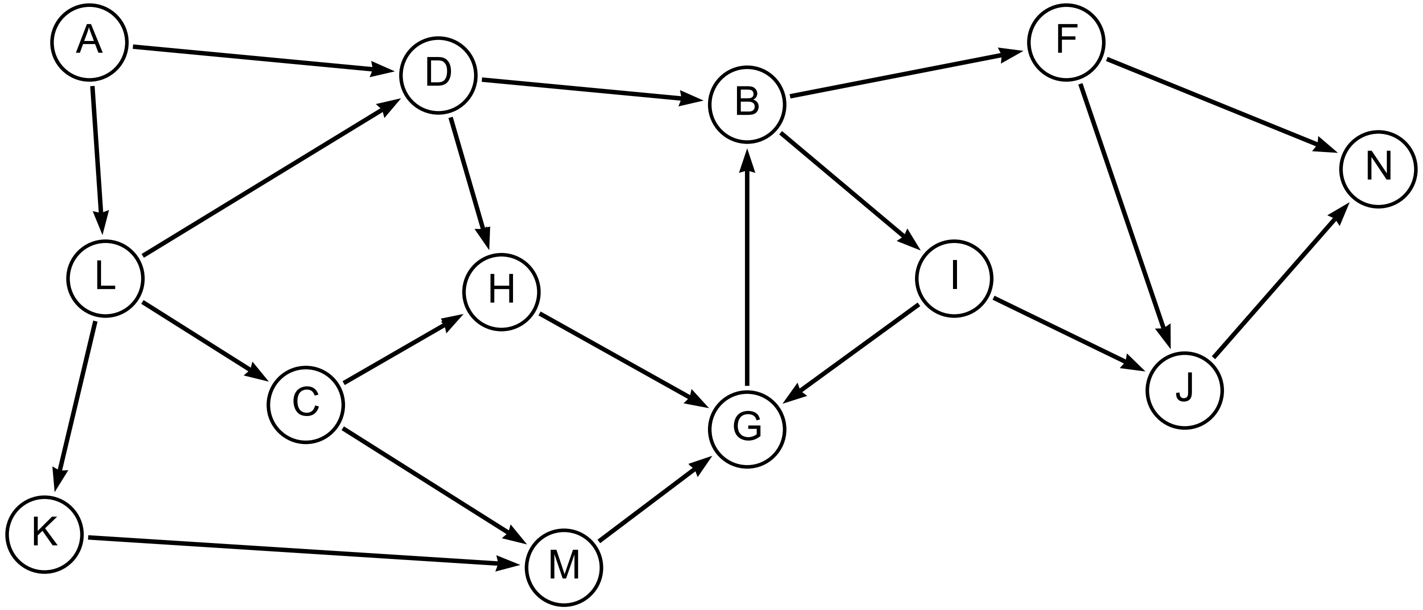
$\forall w \in L_{out}(v)$, $\exists$ a path from v to w



v	$L_{in}(v)$	$L_{out}(v)$
A		M, D, C, K
B	M, D, C, B	
C		M
D		
F	M, D, C, B	N
G	M, D, C, B	B
H	D, C	B, G
I	M, D, C, B	N, G
J	M, D, C, B, F, I	N
K	A	M
L	A	M, D, C, K
M		
N	M, D, C, B	

E. Cohen et al. Reachability and distance queries via 2-hop labels. SODA 2002: 937-946

2-Hop Labeling



v	$L_{in}(v)$	$L_{out}(v)$
A		M, D, C, K
B	M, D, C, B	
C		M
D		
F	M, D, C, B	N
G	M, D, C, B	B
H	D, C	B, G
I	M, D, C, B	N, G
J	M, D, C, B, F, I	N
K	A	M
L	A	M, D, C, K
M		
N	M, D, C, B	

Case 1: $Q(L, M) = True,\ M \in L_{out}(L)$

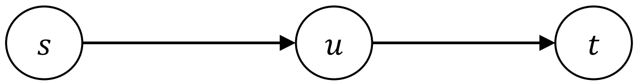
Case 2: $Q(M, B) = True,\ M \in L_{in}(B)$

Case 3: $Q(A, N) = True,\ L_{out}(A) \cap L_{in}(N) \neq \emptyset$

E. Cohen et al. Reachability and distance queries via 2-hop labels. SODA 2002: 937-946

Minimum 2-Hop Labeling

- Index size: $\sum_{v \in V} |L_{in}(v)| + |L_{out}(v)|$
- **Minimum** 2-hop labeling: the index with the minimum index size
  - Intuition: maximally compress the transitive closure
- NP-hard problem [Coh02]
- Efficient heuristics for building 2-hop indexes
  - TFL [Che13], PLL [Aki13], DL [Jin13], and TOL [Zhu14]



Labeling 1

v	$L_{in}(v)$	$L_{out}(v)$
s		u
u		
t	u	

Smaller

Labeling 2

v	$L_{in}(v)$	$L_{out}(v)$
s		
u	s	t
t	s	

Larger

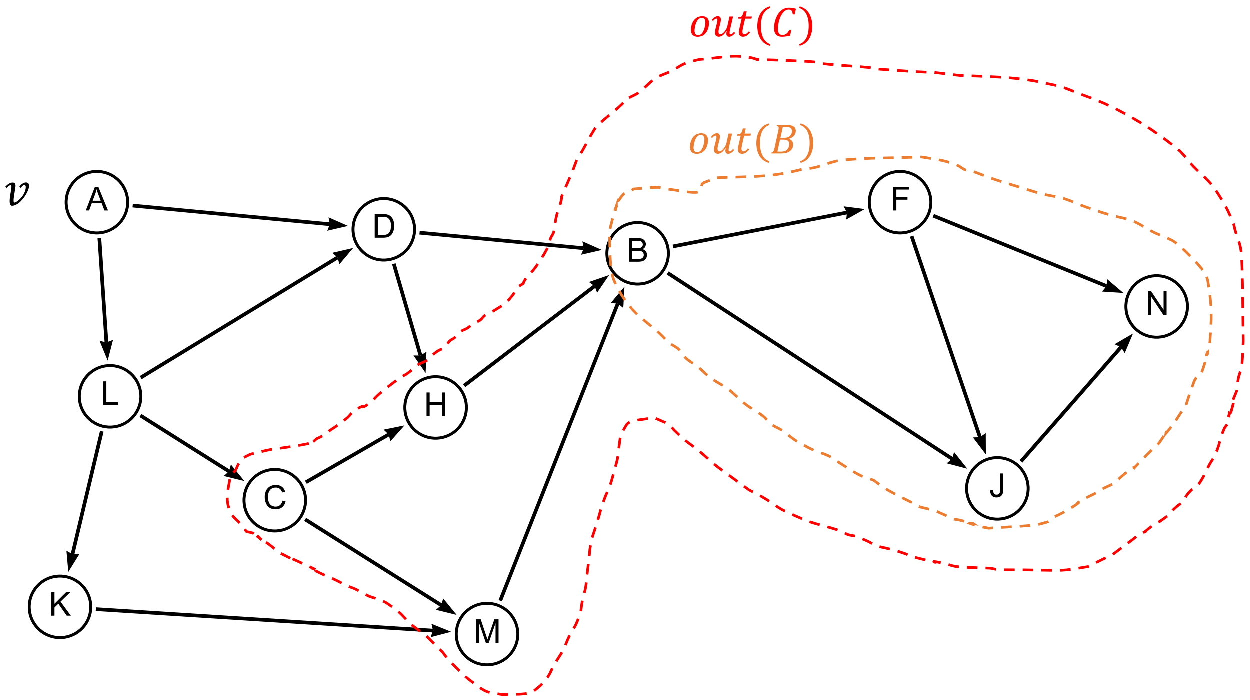
[Coh02] E. Cohen et al. Reachability and distance queries via 2-hop labels. SODA 2002: 937-946
[Che13] J. Cheng et al. TF-Label: a topological-folding labeling scheme for reachability querying in a large graph. SIGMOD Conference 2013: 193-204
[Jin13] R. Jin et al. Simple, Fast, and Scalable Reachability Oracle. Proc. VLDB Endow. 6(14): 1978-1989 (2013)
[Aki13] E. Akiba et al. Fast exact shortest-path distance queries on large networks by pruned landmark labeling. SIGMOD Conference 2013: 349-360
[Zhu14] A. Zhu et al. Reachability queries on large dynamic graphs: a total order approach. SIGMOD Conference 2014: 1323-1334

Outline

1. Plain Reachability Indexes
   a) Tree-Cover Indexes
   b) 2-Hop Indexes
   c) Approximated Transitive Closures
2. Path-Constrained Reachability Indexes
   a) Indexes for Alternation-Based Queries
   b) Indexes for Concatenation-Based Queries
3. Open Challenges

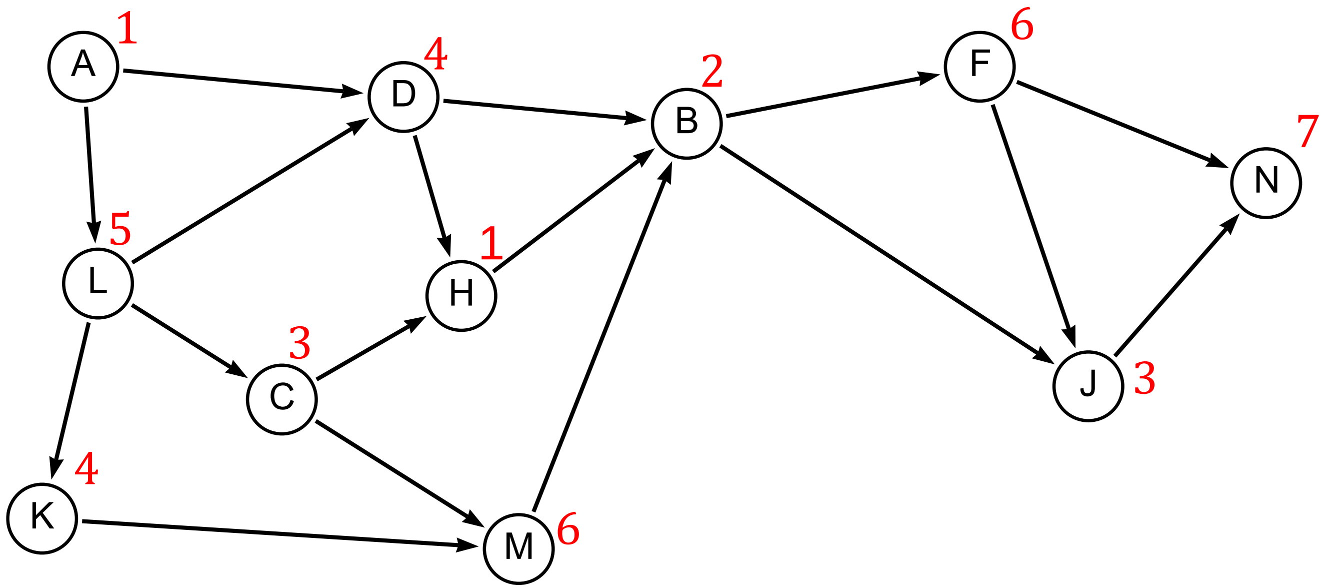
Rethinking of Transitive Closures

- $out(v)$:
  - v and all the vertices that are reachable from v
- **Observation**:
  - *If v is reachable from u, $out(v) \subseteq out(u)$*
  - Example: B is reachable from C
- **Contrapositive condition**:
  - *If $out(v) \nsubseteq out(u)$, v is not reachable from u*
- Computing approximate $out(v)$:
  - K-min-wise independent permutation: IP [Wei14]
  - Bloom filter: BFL [Su17]



[Wei14] H. Wei et al. Reachability Querying: An Independent Permutation Labeling Approach. Proc. VLDB Endow. 7(12): 1191-1202 (2014)
[Su17] J. Su et al. Reachability Querying: Can It Be Even Faster? IEEE Trans. Know. Data Eng. 29(3): 683-697 (2017)

BFL



v	$in(v)$	$out(v)$
A	{1}	{1,2,3,4,5,6,7}
B	{1,2,3,4,5,6}	{2,3,6,7}
C	{1,3,5}	{1,2,3,6,7}
D	{1,4,5}	{1,2,3,4,6,7}
F	{1,2,3,4,5,6}	{6,3,7}
H	{1,3,4,5}	{1,2,3,6,7}
J	{1,2,3,4,5,6}	{3,7}
K	{1,4,5}	{2,3,4,6,7}
L	{1,5}	{1,2,3,4,5,6,7}
M	{1,3,4,5,6}	{2,3,6,7}
N	{1,2,3,4,5,6,7}	{7}

- $Q_r(B, C)$:
  - Index lookup: $out(C) \nsubseteq out(B)$, thus immediately return False
- $Q_r(D, M)$:
  - Index lookup: $out(M) \subseteq out(D)$ and $in(D) \subseteq out(M)$, thus perform guided DFS from D
  - None of the out-neighbors of D can reach M, return False

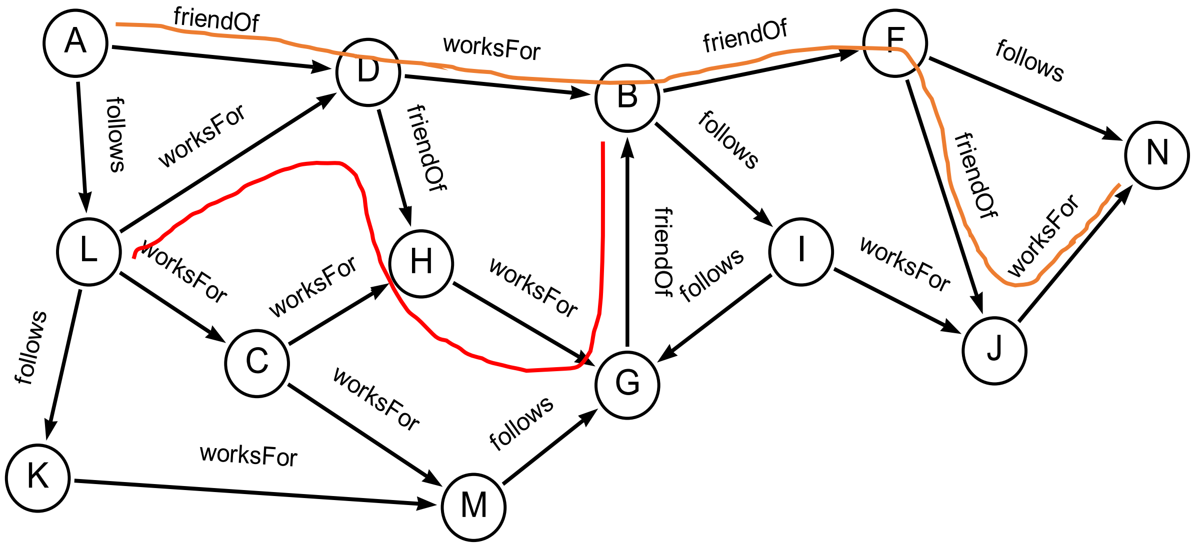
J. Su et al. Reachability Querying: Can It Be Even Faster? IEEE Trans. Know. Data Eng. 29(3): 683-697 (2017)

Outline

1. Plain Reachability Indexes
    a) Tree-Cover Indexes
    b) 2-Hop Indexes
    c) Approximated Transitive Closures
2. Path-Constrained Reachability Indexes
    a) Indexes for Alternation-Based Queries
    b) Indexes for Concatenation-Based path constraints
3. Open Challenges

Path-Constrained Reachability Queries

- $Q_r(s,t,\alpha), \alpha = (\boldsymbol{l_1} \cup \cdots \cup \boldsymbol{l_k})^*$
  - Alternation-based reachability
  - E.g., $Q_r(A, N, (worksFor \cup friendOf)^*)$ = True
- $Q_r(s,t,\alpha), \alpha = (\boldsymbol{l_1} \cdot \ldots \cdot \boldsymbol{l_k})^*$
  - Concatenation-based reachability
  - E.g., $Q_r(L, B, (worksFor \cdot friendOf)^*)$ = True
- Indexes are specifically designed for each type

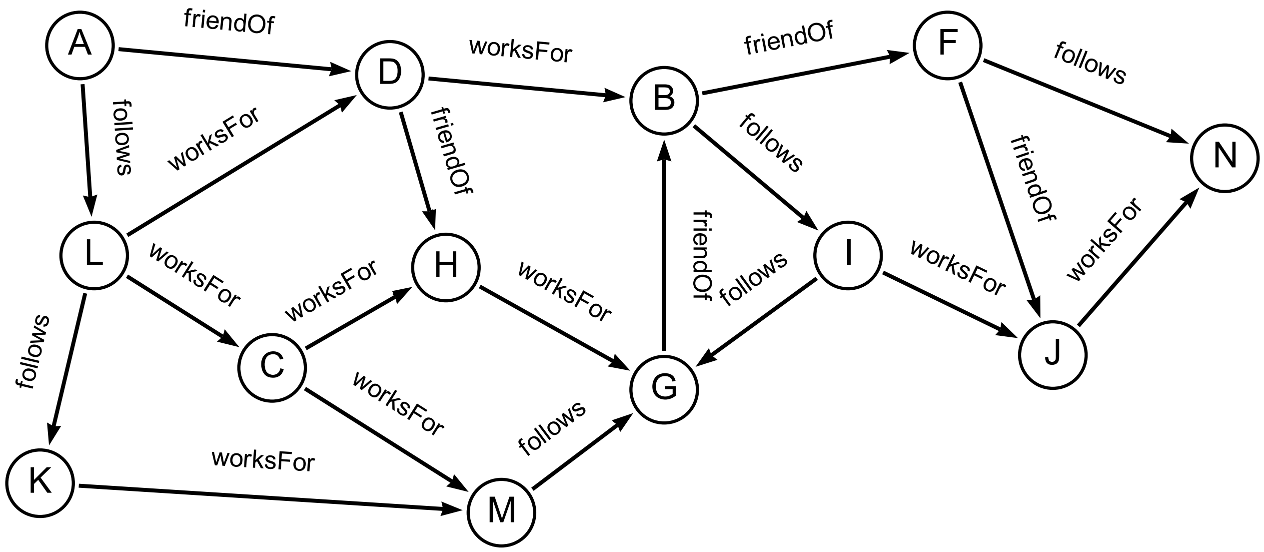


Outline

1. Plain Reachability Indexes
   a) Tree-Cover Indexes
   b) 2-Hop Indexes
   c) Approximated Transitive Closures
2. Path-Constrained Reachability Indexes
   a) Indexes for Alternation-Based Queries
   b) Indexes for Concatenation-Based Queries
3. Open Challenges

Sufficient Path-Label Sets (SPLS)

- Two path-label sets from L to M
  - $\{worksFor, follows\}$
  - $\{worksFor\}$
- $\{worksFor, follows\}$ is redundant
  - $\{worksFor\} \subset \{worksFor, follows\}$
- SPLSs are minimal sets of all path-label sets from a source to a target



R. Jin et al. Computing label-constraint reachability in graph databases. SIGMOD Conference 2010: 123-134

Indexes for Alternation-Based Reachability

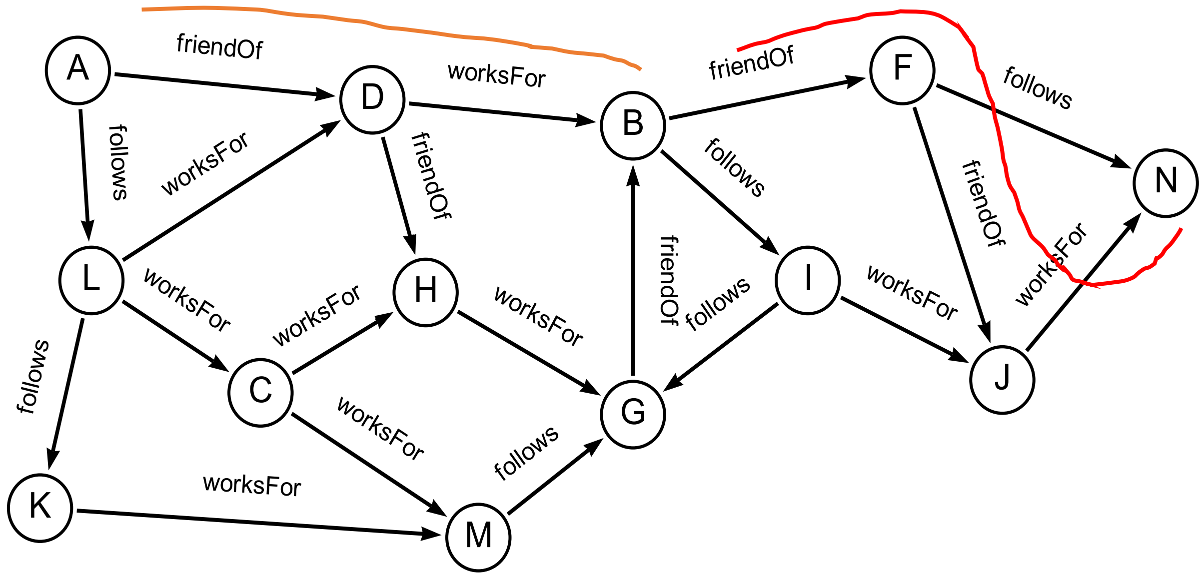
Indexing technique	Framework	Index type	Input	Dynamic	References
Jin et al.	Tree cover	Complete	General Graph	No	[Jin10]
Chen et al.	Tree cover	Complete	General Graph	No	[Che21]
Zou et al.	Generalized TC	Complete	General Graph	Yes	[Xu11,Zou14]
Landmark index	Generalized TC	Partial	General Graph	No	[Val17]
P2H+	2-Hop	Complete	General Graph	No	[Pen20]
DLCR	2-Hop	Complete	General Graph	Yes	[Che22]

Three index frameworks:

- Tree cover
- Generalized TC
- 2-Hop

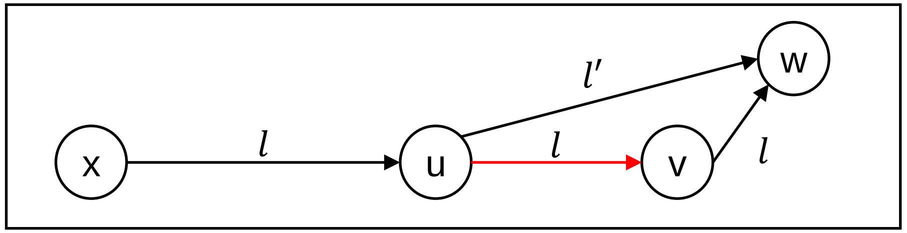
Label-Constrained 2-Hop Labeling

- Intuition of P2H+:
  - Plain reachability is transitive
  - SPLSs are transitive
  - Adding SPLSs into the 2-hop labeling
- $Q_r(A, N, (worksFor \cup friendOf)^*)$:
  - Plain reachability:
    - A can reach B
    - B can reach N
  - Path constraints:
    - SPLSs from A to B contains $\{worksFor, friendOf\}$
    - SPLSs from B to N contains $\{worksFor, friendOf\}$
  - Thus, the answer is True



Dynamic Label Constrained Reachability

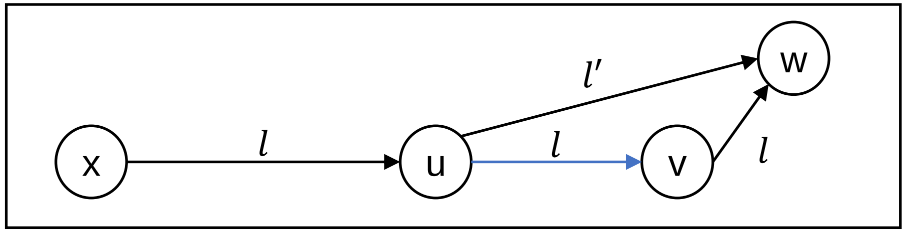
- DLCR: an extension of P2H+ to dynamic graphs
- Inserting (u, v) with label l in DLCR:



Inserting the reachability from x to v

Deleting the **redundant** reachability from x to v with $\{l, l'\}$

- Deleting (u, v) with label l in DLCR:



Deleting the reachability from x to v

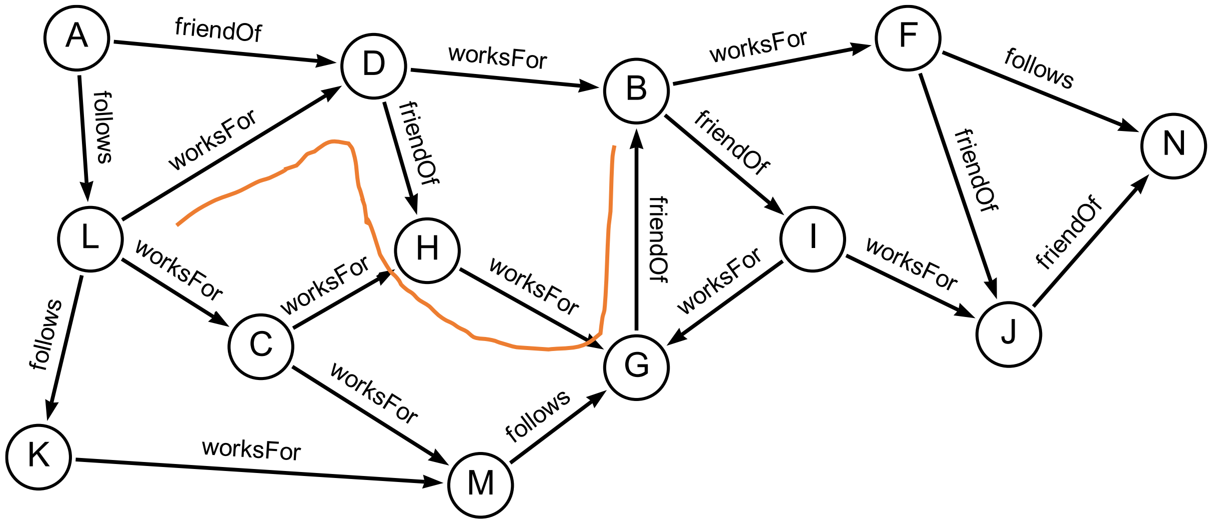
Inserting the **pruned** reachability from x to v with $\{l, l'\}$

X. Chen et al. DLCR: Efficient Indexing for Label-Constrained Reachability Queries on Large Dynamic Graphs. Proc. VLDB Endow. 15(8): 1645-1657 (2022)

Outline

1. Plain Reachability Indexes
   a) Tree-Cover Indexes
   b) 2-Hop Indexes
   c) Approximated Transitive Closures
2. Path-Constrained Reachability Indexes
   a) Indexes for Alternation-Based Queries
   b) Indexes for Concatenation-Based Queries
3. Open Challenges

Minimum Repeats



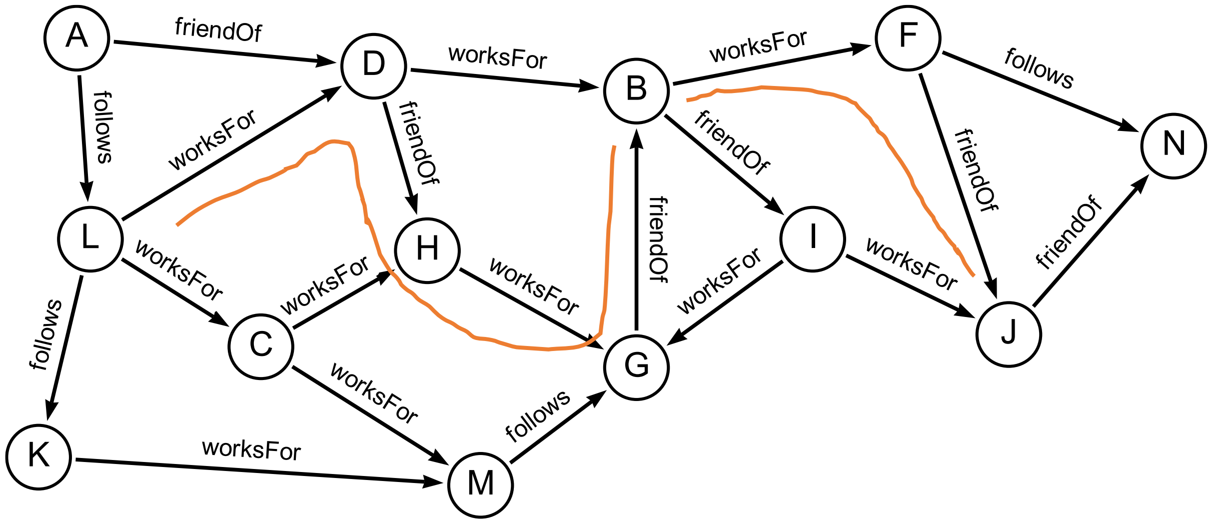
- Efficiently store path-label sequences
  - Minimum repeats of path-label sequences
- Example:
  - Path: $(L, worksFor, D, friendOf, H, worksFor, G, friendOf, B)$
  - Minimum repeat: $(worksFor, friendOf)$

C. Zhang et al. A Reachability Index for Recursive Label-Concatenated Graph Queries. ICDE 2023: 66-80

RLC Index

RLC Index ($k \leq 2$)
(incomplete view)

v	$L_{in}(v)$	$L_{out}(v)$
A	$\ldots$	$(B, (friendOf, worksFor)), \ldots$
B	$\ldots$	$\ldots$
C	$\ldots$	$(G, (worksFor)), \ldots$
D	$\ldots$	$(B, (worksFor)), \ldots$
F	$\ldots$	$\ldots$
G	$(B, (friendOf, worksFor)), \ldots$	$(B, (friendOf)), \ldots$
H	$(L, (worksFor)), \ldots$	$(B, (worksFor, friendOf)), \ldots$
I	$\ldots$	$(B, (worksFor, friendOf)), \ldots$
J	$(B, (worksFor, friendOf))$, $(B, (friendOf, worksFor)), \ldots$	$\ldots$
K	$\ldots$	$\ldots$
L	$\ldots$	$(B, (worksFor, friendOf))$, $(B, (worksFor)), \ldots$
M	$(L, (follows, worksFor)), \ldots$	$(B, (follows, friendOf)), \ldots$
N	$(B, (worksFor, follows)), \ldots$	$\ldots$



Example:

- $Q_r(L, J, \alpha),\ \alpha = (worsFor \cdot friendOf)^*$
  - $(B, (worksFor, friendOf)) \in L_{out}(L)$
  - $(B, (worksFor, friendOf)) \in L_{in}(J)$
  - True

C. Zhang et al. A Reachability Index for Recursive Label-Concatenated Graph Queries. ICDE 2023: 66-80

Outline

An Overview of Main Challenges

- Real-world graphs are
  - large, and
  - fully dynamic
- Plain reachability indexes
  - State-of-the-art indexes can be built efficiently on large graphs
  - Updating indexes is not efficient
- Path-constrained reachability indexes
  - Struggling with both scalability and index updates
  - Indexes for general types of path constraints

References: 1972 - 2013

[Tar72] R. Tarjan. Depth-First Search and Linear Graph Algorithms. SIAM J. Comput. 1(2): 146-160 (1972)

[Agr89] R. Agrawal et al. Efficient Management of Transitive Relationships in Large Data and Knowledge Bases. SIGMOD Conference 1989: 253-262

[Coh02] E. Cohen et al. Reachability and distance queries via 2-hop labels. SODA 2002: 937-946

[Che05] L. Chen et al. Stack-based Algorithms for Pattern Matching on DAGs. VLDB 2005: 493-504

[Sch05] R. Schenkel et al. Efficient creation and incremental maintenance of the HOPI index for complex XML document collections. ICDE 2005: 360-371

[Wan06] H. Wang et al. Dual Labeling: Answering Graph Reachability Queries in Constant Time. ICDE 2006: 75

[Tri07] S. Tril et al. Fast and practical indexing and querying of very large graphs. SIGMOD Conference 2007: 845-856

[Jin08] R. Jin et al. Efficiently answering reachability queries on very large directed graphs. SIGMOD Conference 2008: 595-608

[Jin09] R. Jin et al. 3-HOP: a high-compression indexing scheme for reachability query. SIGMOD Conference 2009: 813-826

[Bra10] R. Bramandia et al. Incremental Maintenance of 2-Hop Labeling of Large Graphs. IEEE Trans. Knowl. Data Eng. 22(5): 682-698 (2010)

[Cai10] J. Cai et al. Path-hop: efficiently indexing large graphs for reachability queries. CIKM 2010: 119-128

[Jin10] R. Jin et al. Computing label-constraint reachability in graph databases. SIGMOD Conference 2010: 123-134

[Yil10] H. Yildirim et al. GRAIL: Scalable Reachability Index for Large Graphs. Proc. VLDB Endow. 3(1): 276-284 (2010)

[Jin11] R. Jin et al. Path-tree: An efficient reachability indexing scheme for large directed graphs. ACM Trans. Database Syst. 36(1): 7:1-7:44 (2011)

[Xu11] K. Xu et al. Answering label-constraint reachability in large graphs. CIKM 2011: 1595-1600

[Che13] J. Cheng et al. TF-Label: a topological-folding labeling scheme for reachability querying in a large graph. SIGMOD Conference 2013: 193-204

[Jin13] R. Jin et al. Simple, Fast, and Scalable Reachability Oracle. Proc. VLDB Endow. 6(14): 1978-1989 (2013)

[Seu13] S. Seufert et al. FERRARI: Flexible and efficient reachability range assignment for graph indexing. ICDE 2013: 1009-1020

[Yan13] Y. Yano et al. Fast and scalable reachability queries on graphs by pruned labeling with landmarks and paths. CIKM 2013: 1601-1606

[Yil13] H. Yildirim et al. DAGGER: A Scalable Index for Reachability Queries in Large Dynamic Graphs. CoRR abs/1301.0977 (2013)

References: 2014 - 2023

[Zou14] L. Zou et al. Efficient processing of label-constraint reachability queries in large graphs. Inf. Syst. 40: 47-66 (2014)

[Zhu14] A. Zhu et al. Reachability queries on large dynamic graphs: a total order approach. SIGMOD Conference 2014: 1323-1334

[Mer14] F. Merz et al. PReaCH: A Fast Lightweight Reachability Index Using Pruning and Contraction Hierarchies. ESA 2014: 701-712

[Vel14] R. Veloso et al. Reachability Queries in Very Large Graphs: A Fast Refined Online Search Approach. EDBT 2014: 511-522

[Wei14] H. Wei et al. Reachability Querying: An Independent Permutation Labeling Approach. Proc. VLDB Endow. 7(12): 1191-1202 (2014)

[Val17] L. Valstar et al. Landmark Indexing for Evaluation of Label-Constrained Reachability Queries. SIGMOD Conference 2017: 345-358

[Su17] J. Su et al. Reachability Querying: Can It Be Even Faster? IEEE Trans. Know. Data Eng. 29(3): 683-697 (2017)

[Wei18] H. Wei et al. Reachability querying: an independent permutation labeling approach. VLDB J. 27(1): 1-26 (2018)

[Pen20] Y. Peng et al. Answering billion-scale label-constrained reachability queries within microsecond. Proc. VLDB Endow. 13(6): 812-825 (2020)

[Che21] Y. Chen et al. Graph Indexing for Efficient Evaluation of Label-constrained Reachability Queries. ACM Trans. Database Syst. 46(2): 8:1-8:50 (2021)

[Han21] K. Han et. O'Reach: Even Faster Reachability in Large Graphs. SEA 2021: 13:1-13:24

[Lyu21] Q. Lyu et al. DBL: Efficient Reachability Queries on Dynamic Graphs. DASFAA (2) 2021: 761-777

[Che22] X. Chen et al. DLCR: Efficient Indexing for Label-Constrained Reachability Queries on Large Dynamic Graphs. Proc. VLDB Endow. 15(8): 1645-1657 (2022)

[Zha23] C. Zhang et al. A Reachability Index for Recursive Label-Concatenated Graph Queries. ICDE 2023: 66-80