Progress and Challenges for Labeling Schemes

Cyril Gavoille
(LaBRI, University of Bordeaux)

Advances in Distributed Graph Algorithms (ADGA)

Salvador, Brazil - October 19, 2012

Information & Locality

Understanding what information are needed to achieve a computational task is a central question not only in DC (eg., data-structure theory, communication complexity,...)

The ultimate goal in **Labeling Schemes** is to understand how localized and how much information are required to solve a given task on a network.

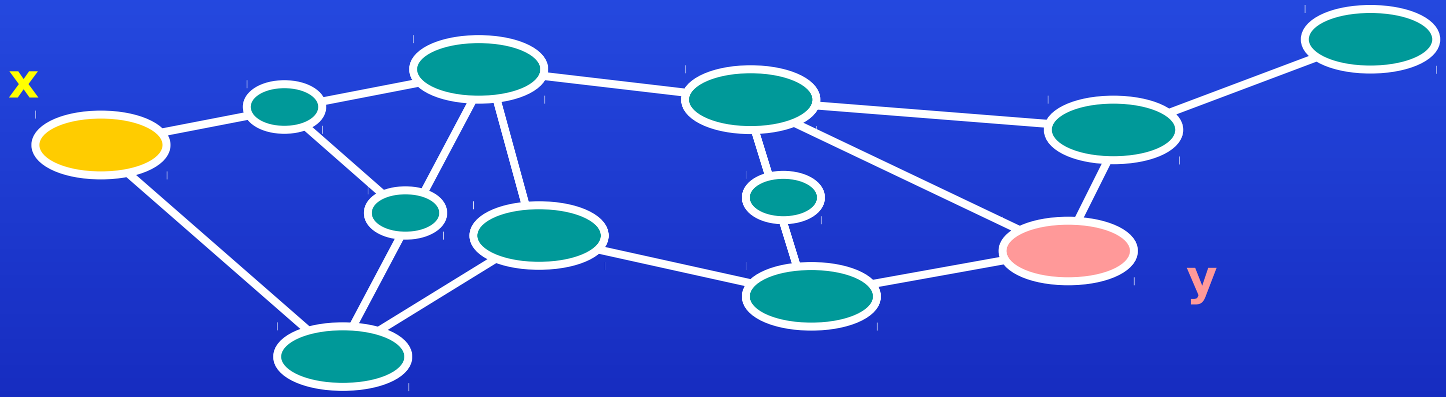
Agenda

1. Informative labeling schemes
2. Forbidden-set labeling schemes
3. Challenges

Agenda

1. Informative labeling schemes
2. Forbidden-set labeling schemes
3. Challenges

Task1: Routing in a physical network



Routing query: next hop to go from x to y?

- pre-processing to compute routing information
- a node x stores only routing information involving x

$\Rightarrow$ distributed data-structure

Task2: Ancestry in rooted trees

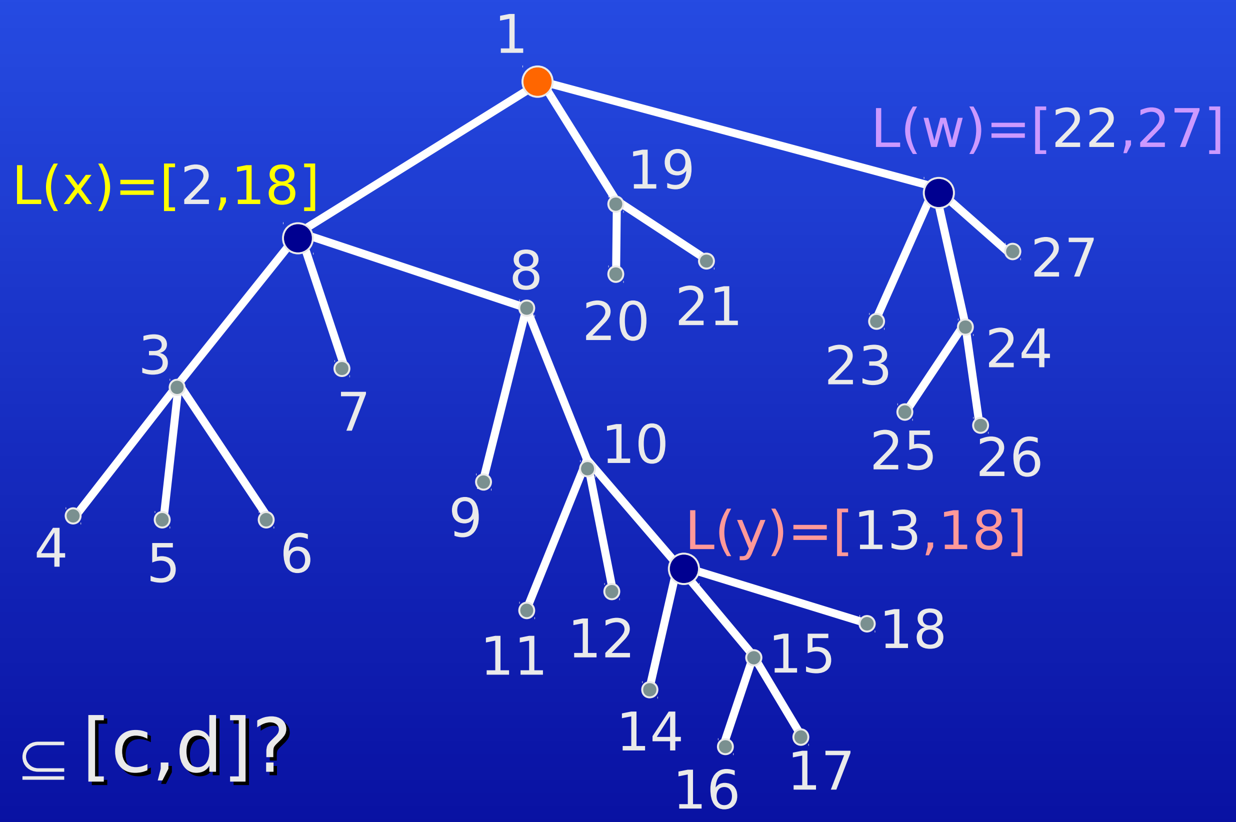
Motivation: [Abiteboul,Kaplan,Milo ’01]

The <TAG> ... </TAG> structure of a huge XML data-base is a rooted tree. Some queries are ancestry relations in this tree.

Ex: Is <“distributed computing”> descendant of <booktitle>?

Use compact index for fast query XML search engine. Here the constants do matter. Saving 1 byte of fast memory on each entry of the index table is important. Here n is large, ~ 10^9.

Folklore solution: DFS labeling



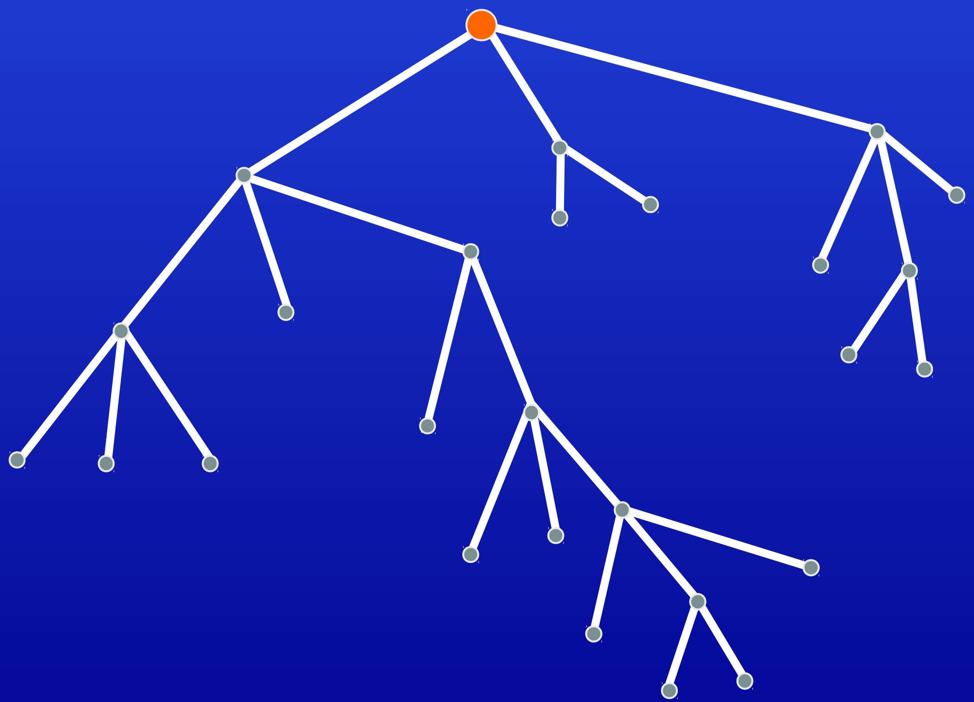
[a,b] $\subseteq$ [c,d]?

$\Rightarrow$ 2logn bit labels

Best solution:
$\log n + \theta(\log\log n)$ bit labels

[Alstrup,Rauhe – Siam J.Comp. ’06]

[Fraigniaud,Korman – STOC’10]



Informative Labeling Schemes
(more formally) [Peleg '00]

Let P be a graph property defined on pairs of vertices (can be extended to tuples), and let F be a graph family.

A P -labeling scheme for F is a pair ‹L,f› such that: $\forall$ G $\in$ F, $\forall$u,v $\in$ G:

- (labeling) L(u,G) is a binary string
- (decoder) f(L(u,G),L(v,G)) = P (u,v,G)

A distributed data-structure

- Get the labels of nodes involved in the query
- Compute/decode the answer from the labels
- No other source of information is required

Some P-labeling schemes

- Adjacency
- Distance (exact or approximate)
- First edge on a (near) shortest path
- Ancestry, parent, nca, sibling relations in trees
- Edge/vertex connectivity, flow
- Proof labeling systems
- ...

Adjacency Labeling
Implicit Representation

$P(x,y,G)$ is true iff $xy \in E(G)$

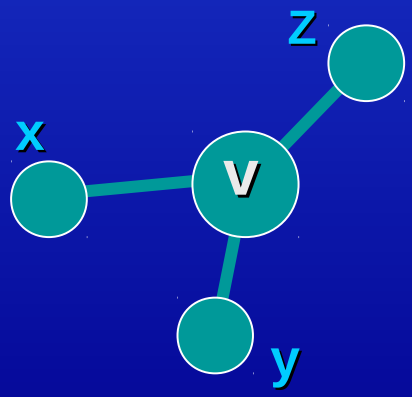
[Kanan,Naor,Rudich – STOC '92]

O(logn) bit labels for:

- trees (and forests)
- bounded arboricity graphs (planar, ...)
- bounded treewidth graphs

In particular:

- 2logn bit labels for trees
- 4logn bit labels for planar

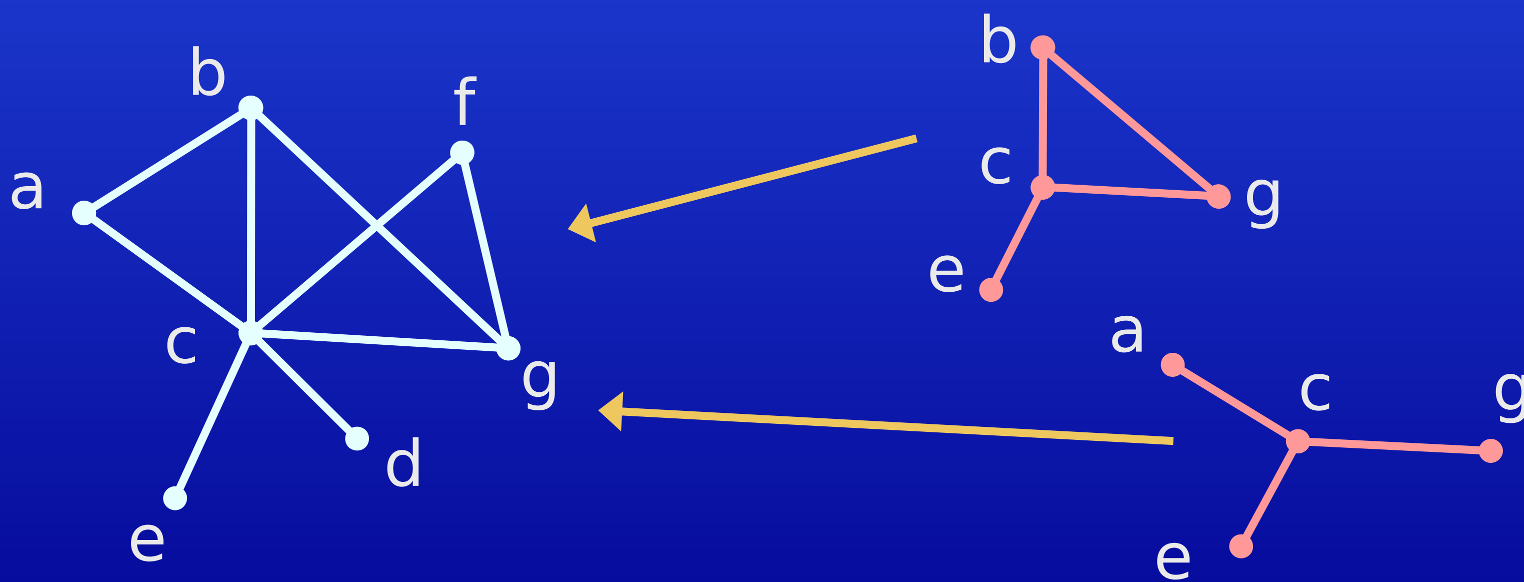


Acutally, the problem is equivalent to an old combinatorial problem:

[Babai,Chung,Erdös,Graham,Spencer '82]

Small Induced-Universal Graph

U is an induced-universal graph for the family F if every graph of F is isomorphic to an induced subgraph of U

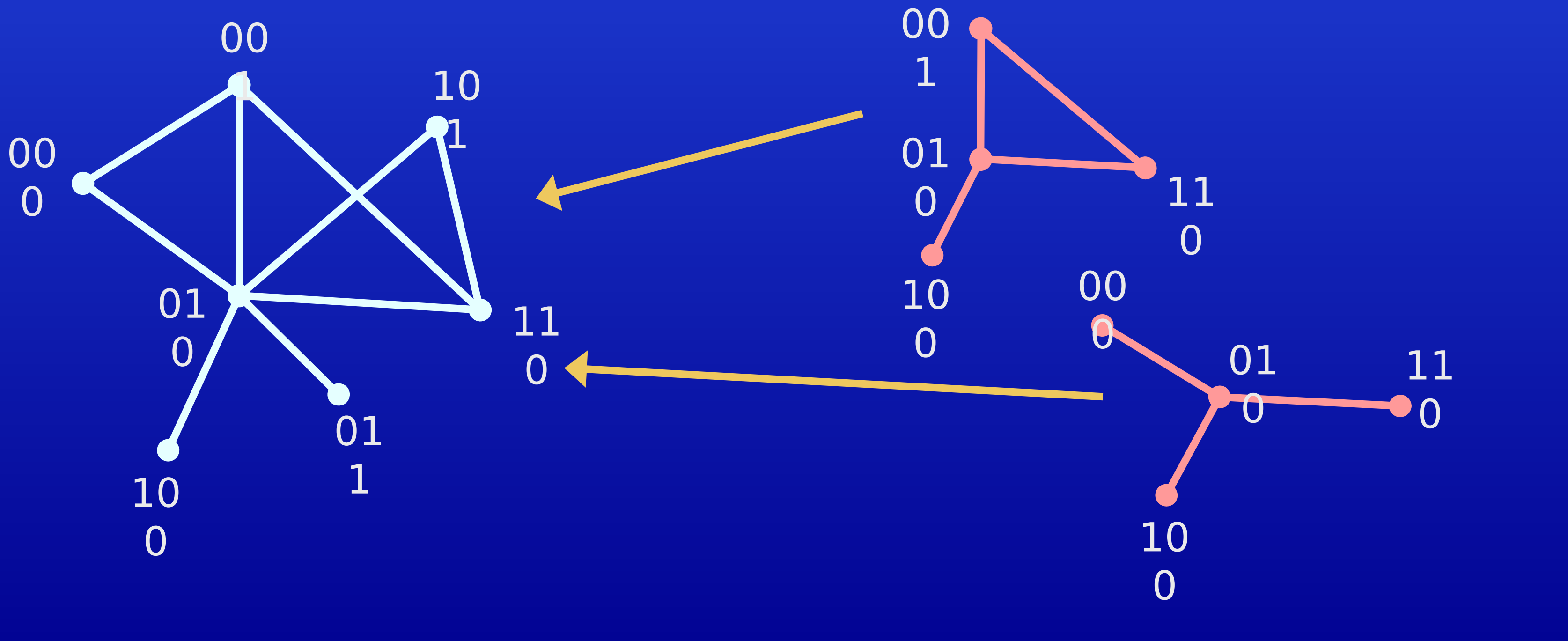


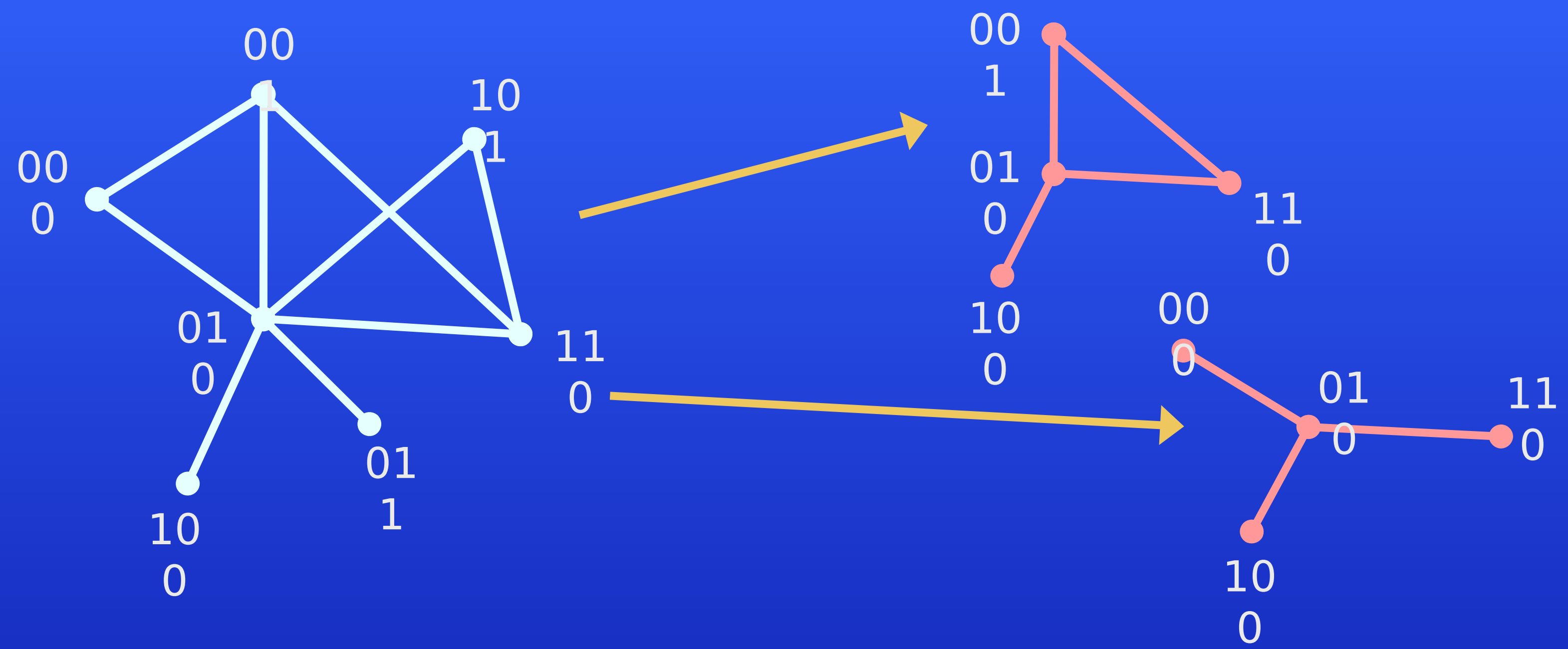
Acutally, the problem is equivalent to an old combinatorial problem:

[Babai,Chung,Erdös,Graham,Spencer ’82]

Small Induced-Universal Graph

U is an induced-universal graph for the family F if every graph of F is isomorphic to an induced subgraph of U





Universal graph U
(fixed for F)

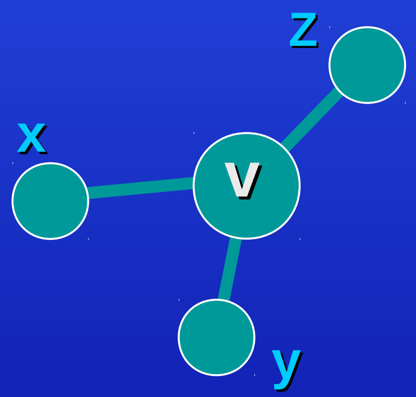
Graph G of F

size of L(x,G) = $\lceil \log_2 |V(U)| \rceil$

Best known results & open questions

- Cubic graphs: $^5/_3 \log n + O(\log\log n)$
  [Esperet,Ochem,Labourel '08]

- Trees: $\log n + O(\log^* n)$
  [Alstrup,Rauhe – FOCS'02]
  $\Rightarrow$ Planar: $3\log n + O(\log^* n)$

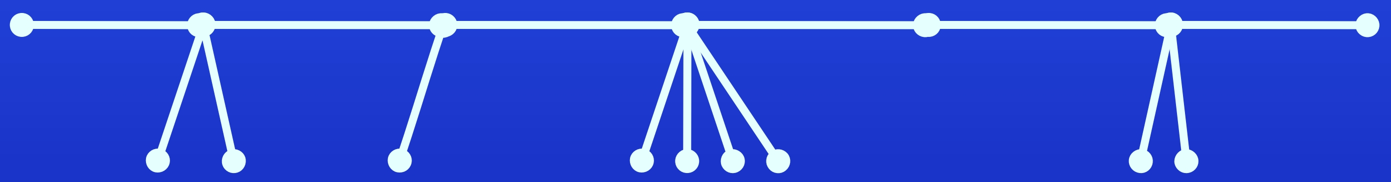


- Planar (minor-free): $2\log n + O(\log\log n)$
  [Gavoille,Labourel – ESA'07]

$\log^* n = \min\{\, i \geq 0 \mid \log^{(i)} n \leq 1 \,\}$

- Lower bounds?: logn + $\Omega(1)$ for planar

logn + O(1) bits for this family?



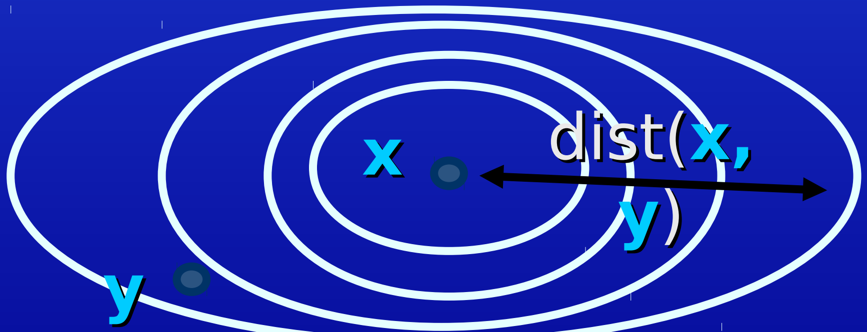
- No hereditary family with $n!2^{O(n)}$ labeled graphs (trees, planar, bounded genus, bounded treewidth, …) is known to require labels of logn + $\omega(1)$ bits

Distance labeling

$P(x,y,G)=dist_G(x,y)$

Motivation: [Peleg '99]

If a short label (say of polylogarithmic size) can be added to the address of the destination, then routing to any destination can be done without routing tables and with a “limited” number of messages.



message header=hop-count

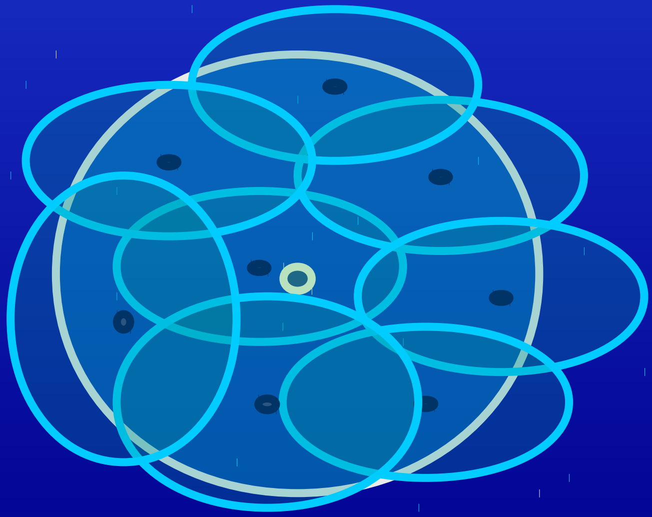
Selected results
(unweighted graphs)

- $\Theta(n)$ bits for general graphs
  - 1.56n bits, but with O(n) time decoder!
    [Winkler ’83 (Squashed Cube Conjecture)]
  - 11n bits and O(loglogn) time decoder
    [Gavoille,Peleg,Pérennès,Raz ’01]
- $\Theta(\log^2 n)$ bits for trees and bounded treewidth graphs, … [Peleg ’99, GPPR ’01]
- $\Theta(\log n)$ bits and O(1) time decoder for interval, permutation graphs, … [ESA’03]: $\Rightarrow$ O(n) space in total O(1) query time, even for $m=\Omega(n^2)$

Results (cont'd)

- $\Theta(\log n \cdot \log\log n)$ bits and $(1+o(1))$-approximation for trees and bounded treewidth graphs [GKKPP – ESA'01]
- Doubling dimension-α graphs

Every radius-2r ball can be covered by $\leq 2^{\alpha}$ radius-r balls



- Euclidean graphs have $\alpha = O(1)$
- Include bounded growing graphs
- Robust notion

Distance labeling for doubling dimension-α graphs

$O(\varepsilon^{-O(\alpha)} \log n \cdot \log\log n)$ bits

$(1+\varepsilon)$-approximation for doubling dimension-α graphs

[Gupta,Krauthgamer,Lee – FOCS’03]

[Talwar – STOC’04]

[Mendel,Har-Peled – SoCG’05]

[Slivkins – PODC’05]

Distance labeling for planar

- $O(\log^2 n)$ bits for 3-approximation

  [Gupta,Kumar,Rastogi – Siam J.Comp '05]
- $O(\varepsilon^{-1}\log^2 n)$ bits for $(1+\varepsilon)$-approximation

  [Thorup – J.ACM '04]

- $\Omega(n^{1/3}) \leq ? \leq \tilde{O}(\sqrt{n})$ for exact distance

- $O(\varepsilon^{-1}\log^2 n)$ bits for $(1+\varepsilon)$-approximation for graphs excluding a fixed minor $(K_5, K_6, \ldots)$

  [Abraham,Gavoille – PODC'06]

Agenda

1. Informative labeling schemes
2. Forbidden-set labeling schemes
3. Challenges

Forbidden-set labeling scheme
(extension of labeling scheme)

- Objectif: to treat more interesting queries

  Given (u,v,w): is there a path from u to v in G\{w}?

  [This particular problem reduces to a classical labeling scheme in bi-component tree (nca & routing) with O(logn) bit labels.]

- Challenge: Given $(u,v,w_1,\ldots,w_k)$:

  Is there a path from u to v in $G\setminus\{w_1,\ldots,w_k\}$?

Emergency planning for connectivity

[Patrascu,Thorup - FOCS'07]

- Motivation: parallel attack (link failure in IP blackbone, earthquake on road network, malicious attack from vorms or viruses,…)

- Con(u,v) $\Rightarrow$ constant time (after pre-processing G)
- Con(u,v,w) $\Rightarrow$ constant time (after pre-proc. G)

- Con(u,v,w_1,…,w_k) $\Rightarrow$ O(k) or Õ(k) time? (after pre-proc. G), and constant time? (after pre-proc. w_1…w_k)

- Note: O(n+m) time is too much. Need a query time depending only on the #nodes involved in the query.

Forbidden-set labeling scheme

[Courcelle,Twigg - STACS'07]

A P –forbidden-set labeling scheme for F is a pair $\langle L,f \rangle$ s.t. $\forall\, G \in F$, $\forall u,v \in G$, $\forall\, X \subseteq G$:

- $L(u,G)$ is a binary string
- $f(L(u,G),L(v,G),L(X,G)) = P\,(u,v,X,G)$

where $L(X,G) := \{L(w,G) : w \in X\}$

Forbidden-set connectivity

[Courcelle,G.,Kanté,Twigg – TGGT’08]

[Borradaile,Pettie,Wulff-Nilsen – SWAT’12]

Connectivity in planar graphs: O(logn) bit labels

[O(loglogn) query time after O(klogk) time for query pre-processing]

Meta-Theorem: [Courcelle,Twigg – STACS’07]

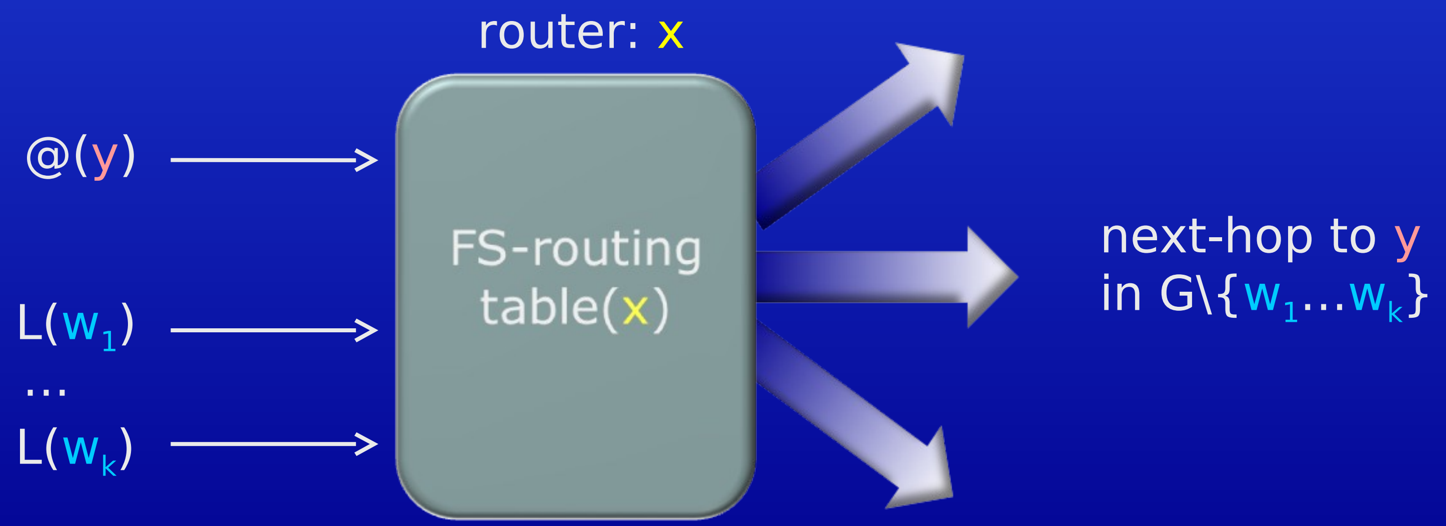
If G has “clique-width” at most cw (generalization of tree-width) and every Monadic Second Order (MSO-logic) predicate P (distances, connectivity, …) then labels of $O(cw^2 \log^2 n)$-bit suffice.

Notes: same (optimal) bounds for distances in trees for the static case, but do not include planar …

Routing with forbidden-sets

Design a routing scheme for G s.t. for every subset X of “forbidden” nodes (crashes, malicious, …) routing tables can be updated efficiently provided X.

⇒ This capture routing policies



Some results for FS routing

[Courcelle,Twigg – STCAS'07]

Clique-width cw: $O(cw^2\log^2 n)$ bit labels and routing tables for shortest path routing.

[Abraham,Chechik,G.,Peleg – PODC'10]

Doubling dimension-α: $O(1+\varepsilon^{-1})^{2\alpha}\log^2 n$ bit labels and routing tables for stretch $1+\varepsilon$ routing (wrt. shortest path)

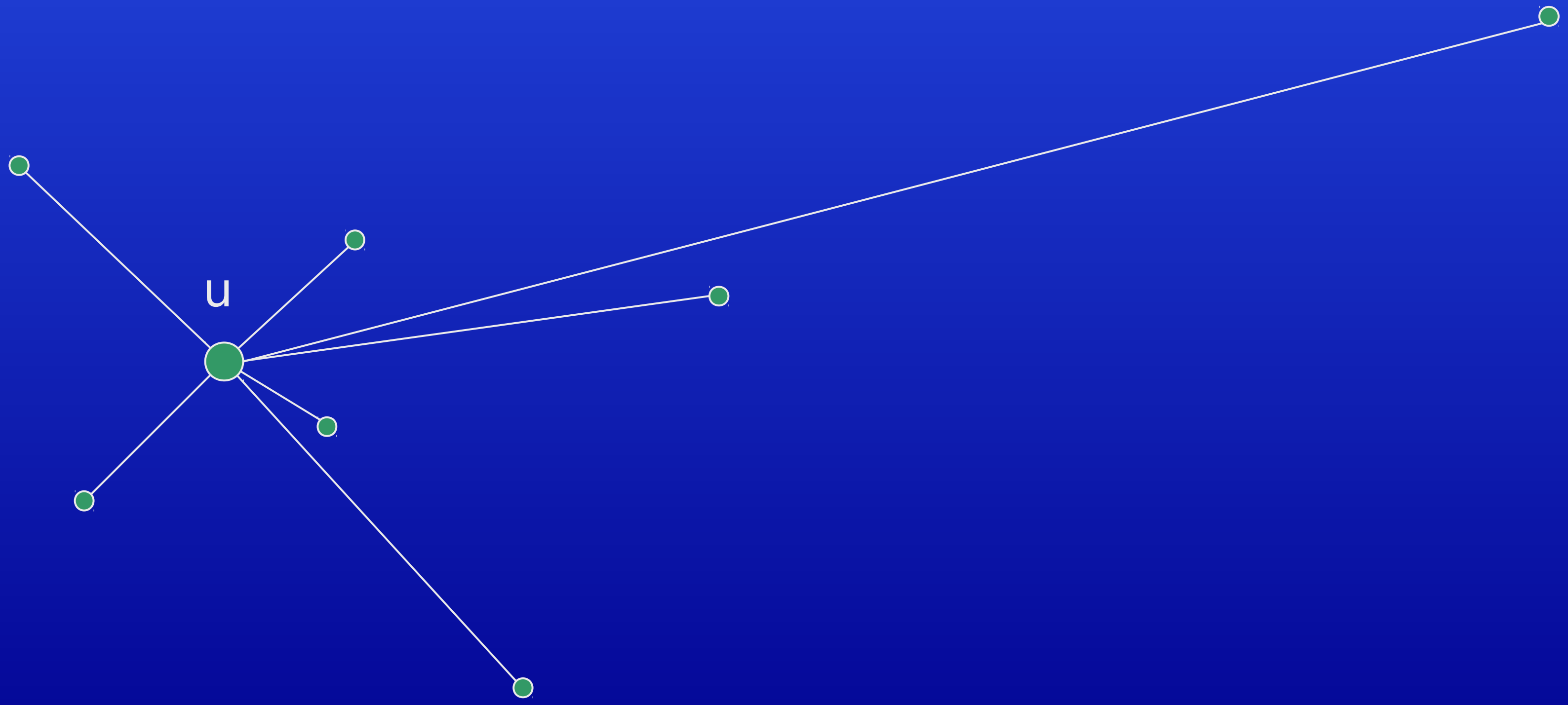
[Abraham,Chechik,G. – STOC'12]

Planar: $O(\varepsilon^{-1}\log^3 n)$ bit routing tables and $O(\varepsilon^{-2}\log^5 n)$ bit labels for stretch $1+\varepsilon$ routing

How does it work?

Query: dist(u,v) in $G \setminus \{w_1 \ldots w_k\}$ given $L(u), L(v), L(w_1), \ldots, L(w_k)$

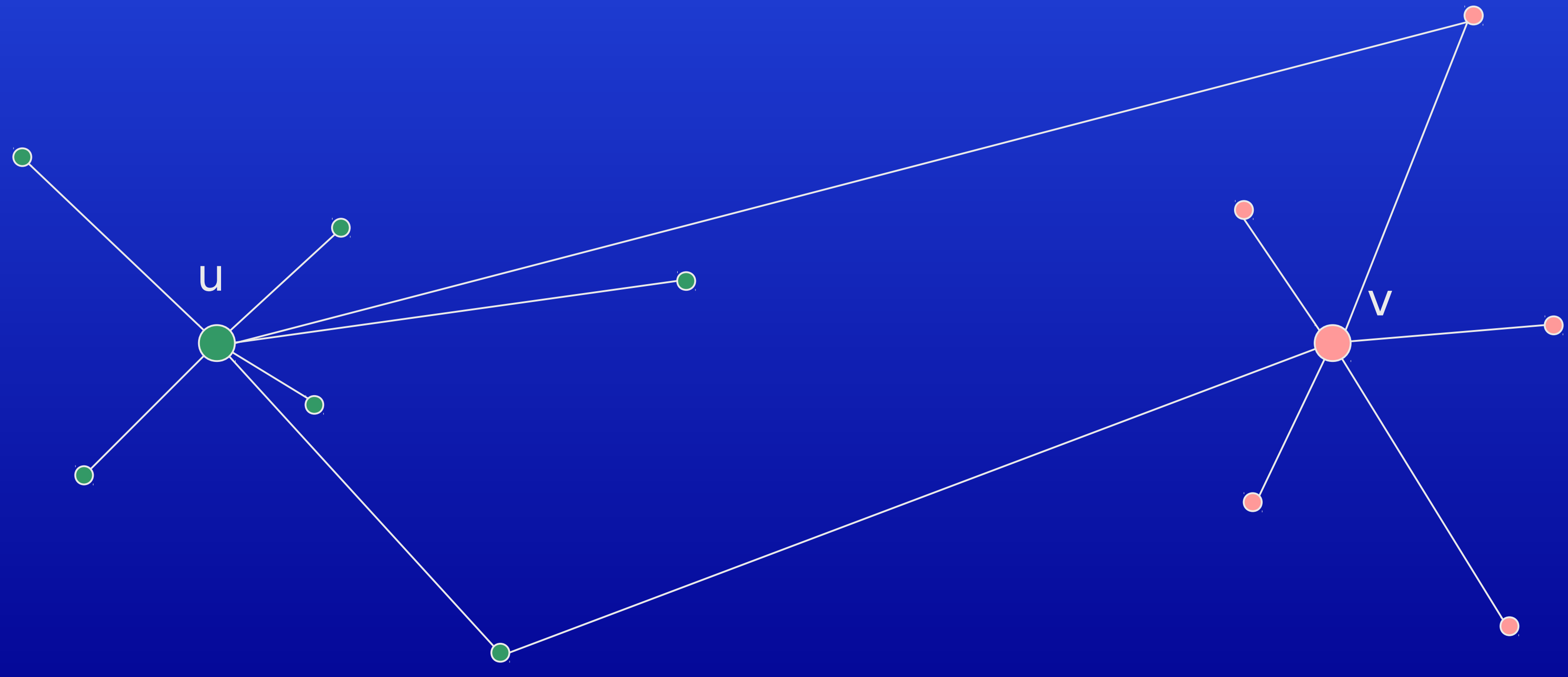
1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size $\tilde{O}(k)$



How does it work?

Query: dist(u,v) in G\{w_1…w_k} given L(u),L(v),L(w_1),…,L(w_k)

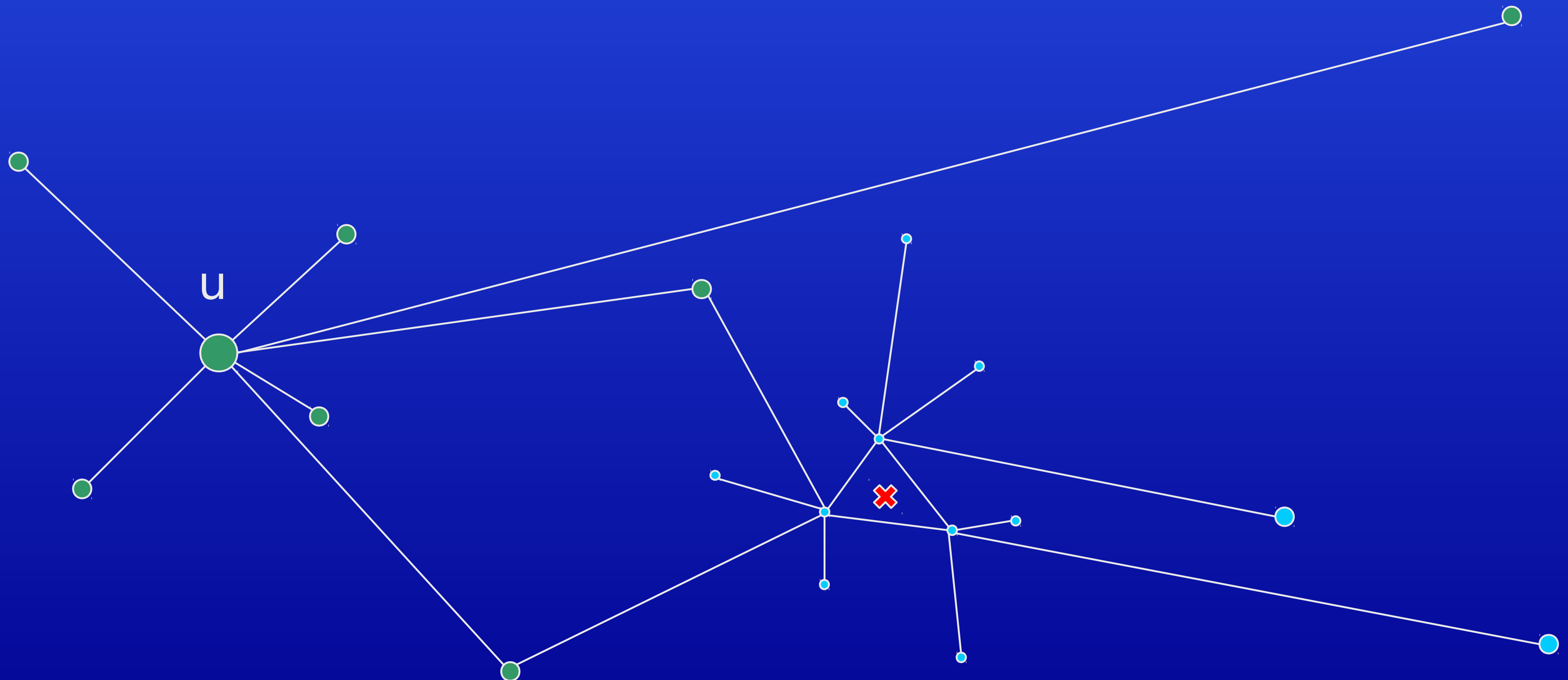
1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size Õ(k)



How does it work?

Query: dist(u,v) in $G\setminus\{w_1 \ldots w_k\}$ given $L(u),L(v),L(w_1),\ldots,L(w_k)$

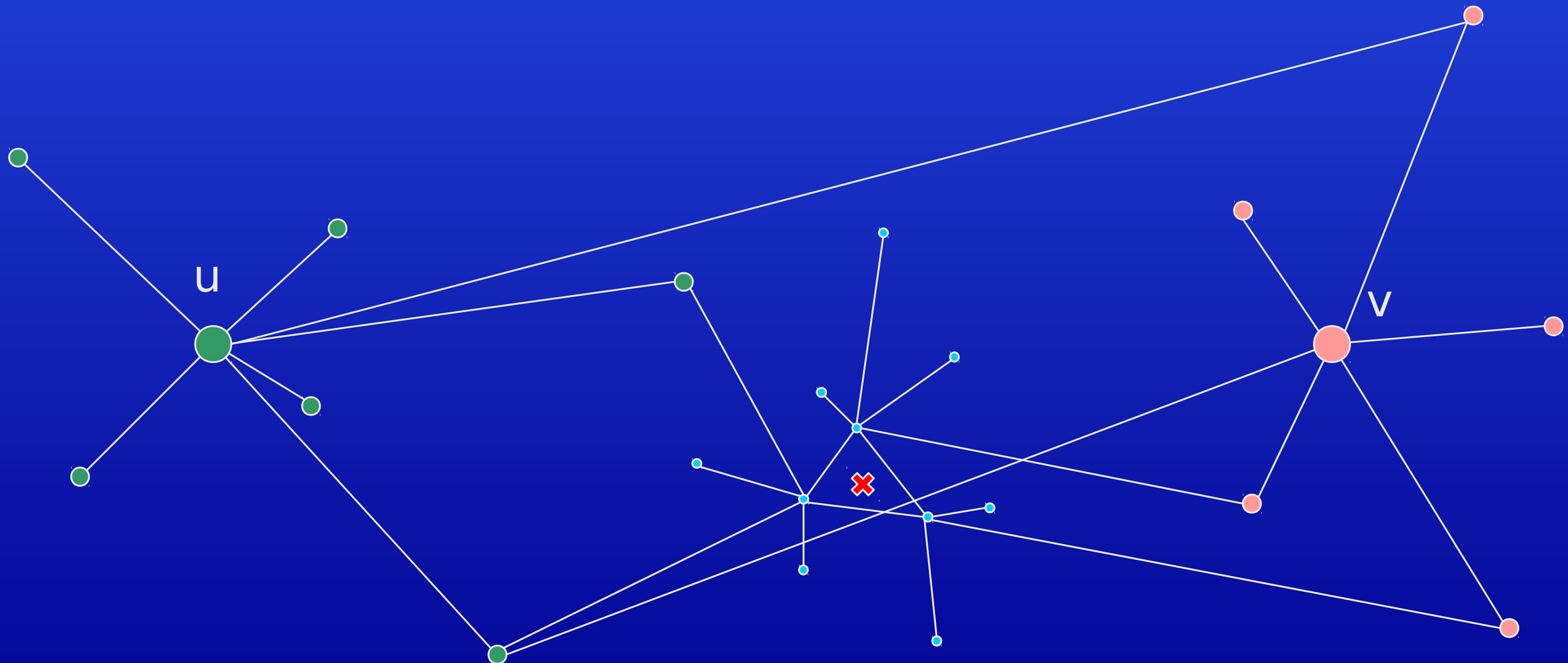
1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size $\tilde{O}(k)$



How does it work?

Query: dist(u,v) in G\{w_1…w_k} given L(u),L(v),L(w_1),…,L(w_k)

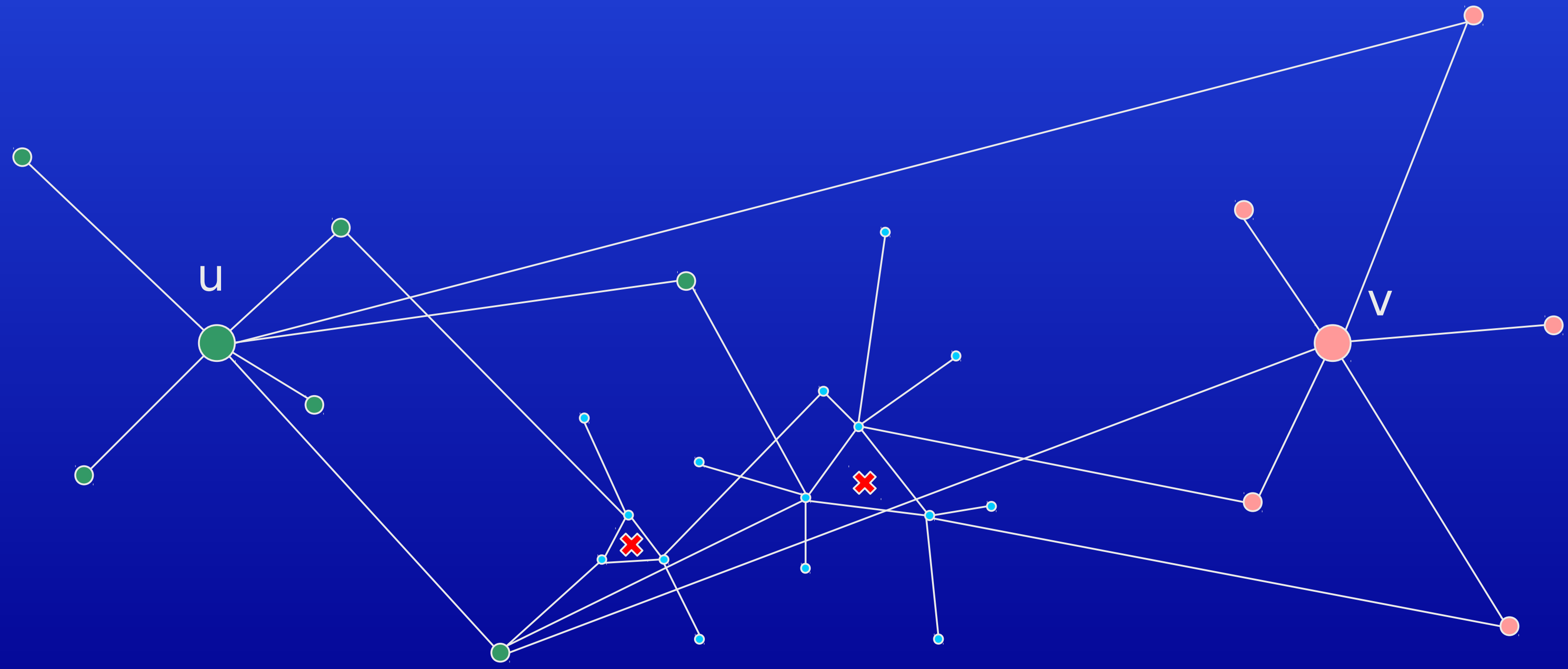
1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size Õ(k)



How does it work?

Query: dist(u,v) in G\{w_1…w_k} given L(u),L(v),L(w_1),…,L(w_k)

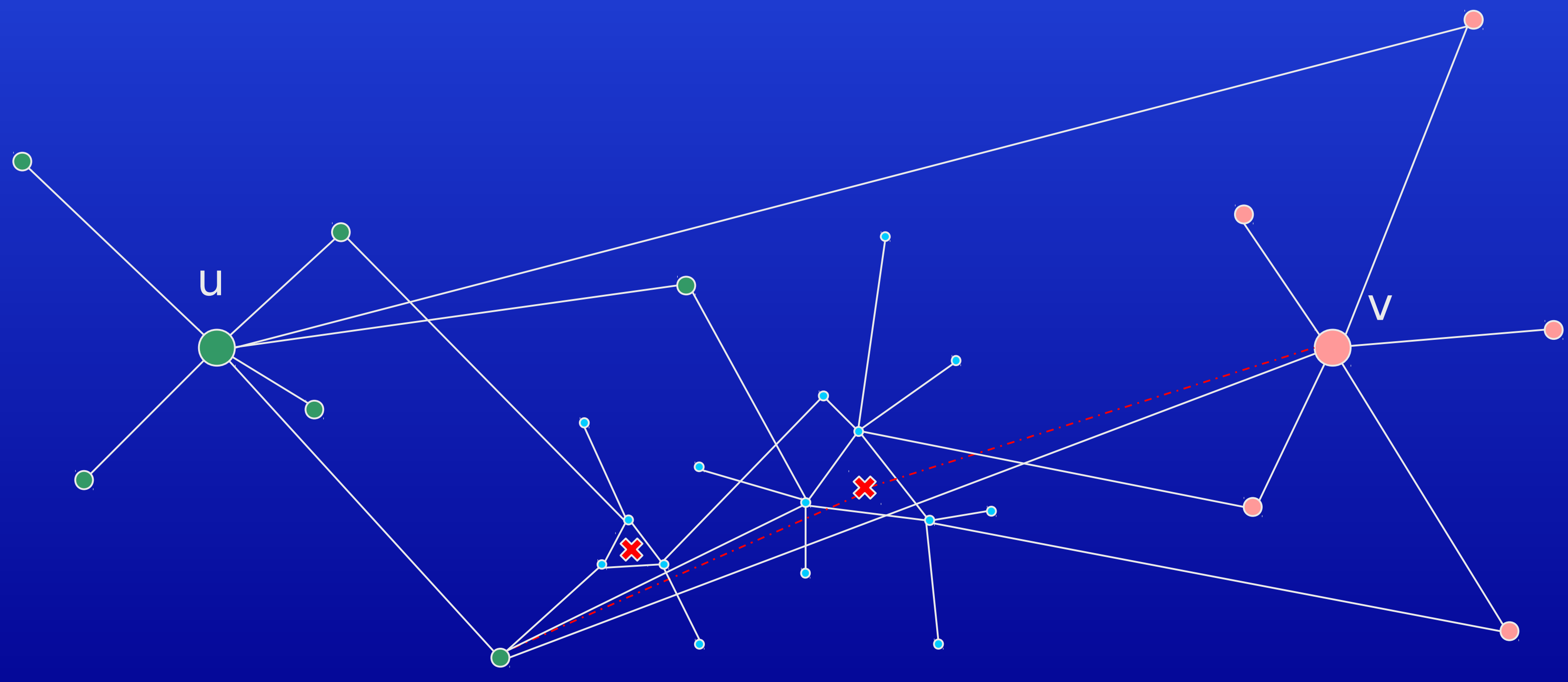
1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size Õ(k)



How does it work?

Query: dist(u,v) in G\{w_1…w_k} given L(u),L(v),L(w_1),…,L(w_k)

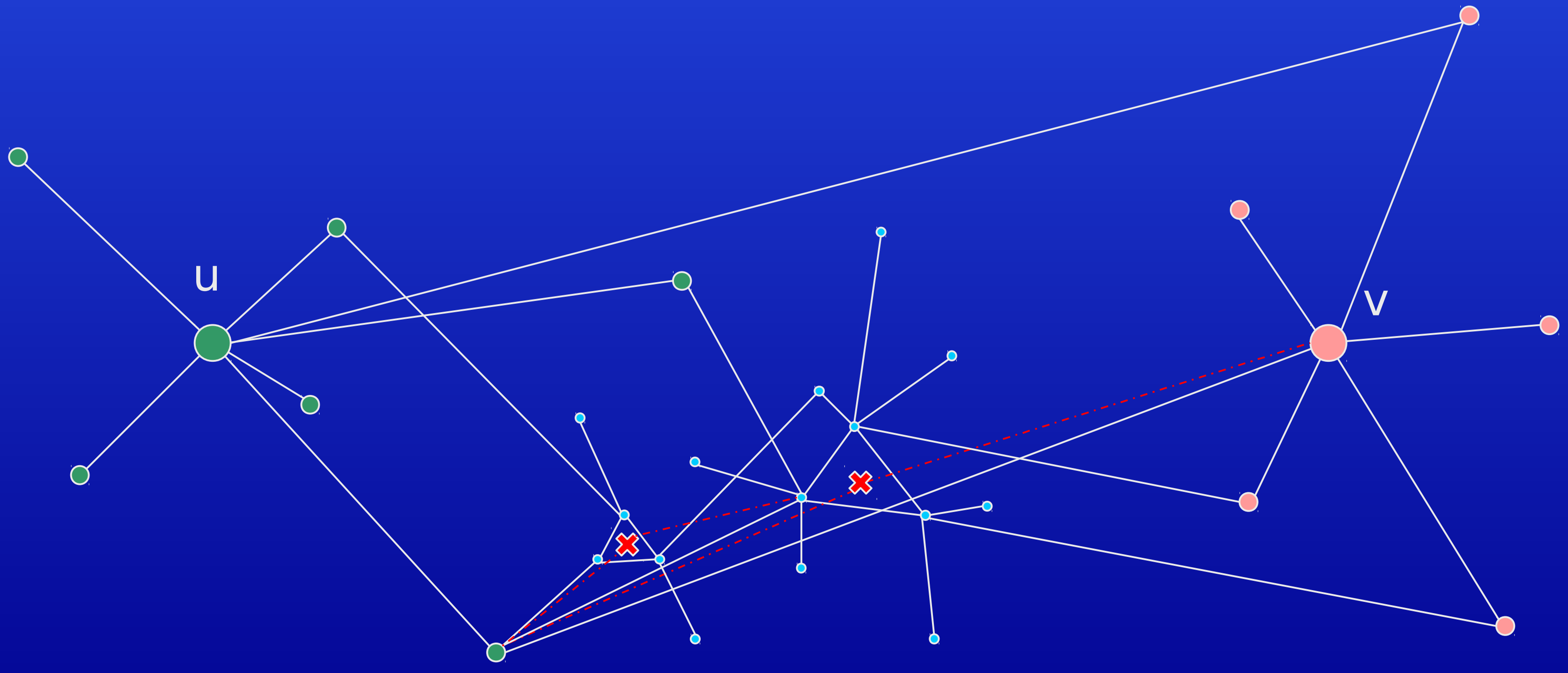
1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size Õ(k)



How does it work?

Query: dist(u,v) in G\{w_1…w_k} given L(u),L(v),L(w_1),…,L(w_k)

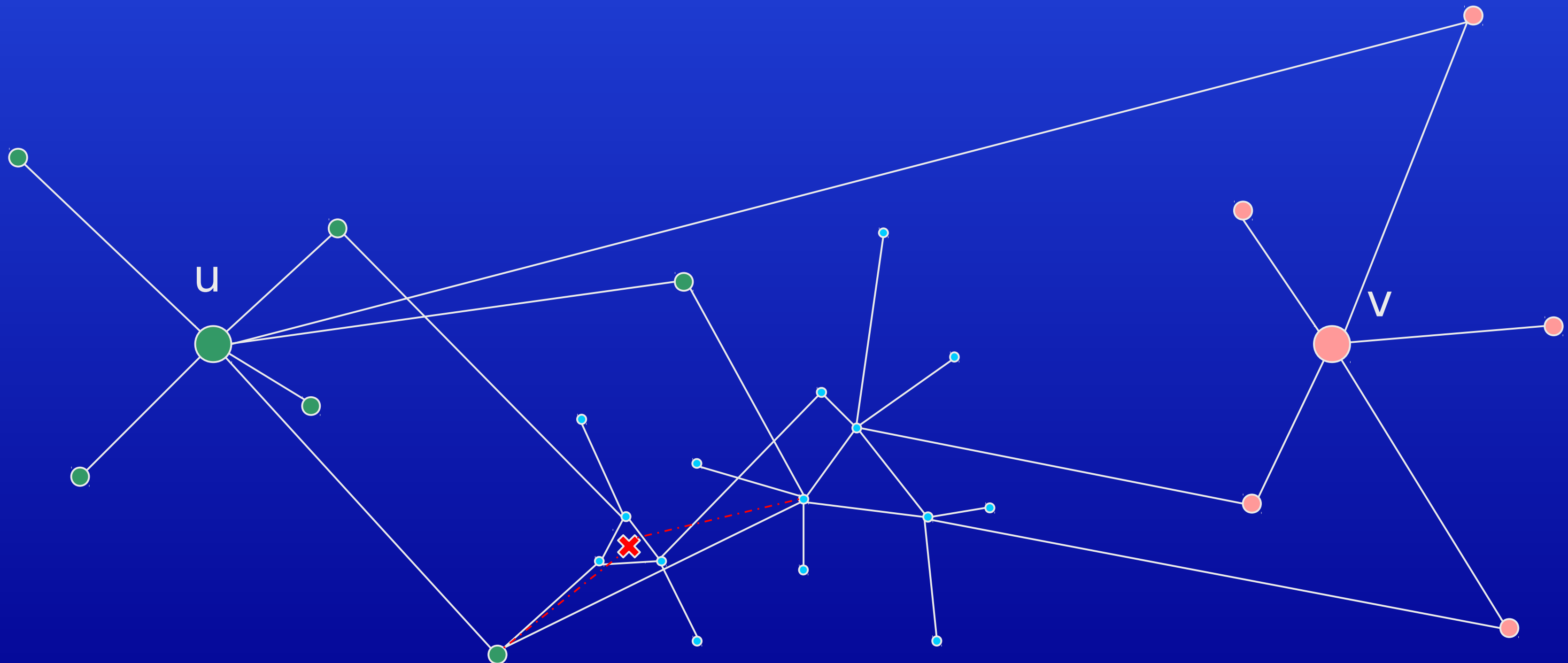
1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size Õ(k)



How does it work?

Query: dist(u,v) in $G\setminus\{w_1 \ldots w_k\}$ given $L(u), L(v), L(w_1), \ldots, L(w_k)$

1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size $\tilde{O}(k)$



How does it work?

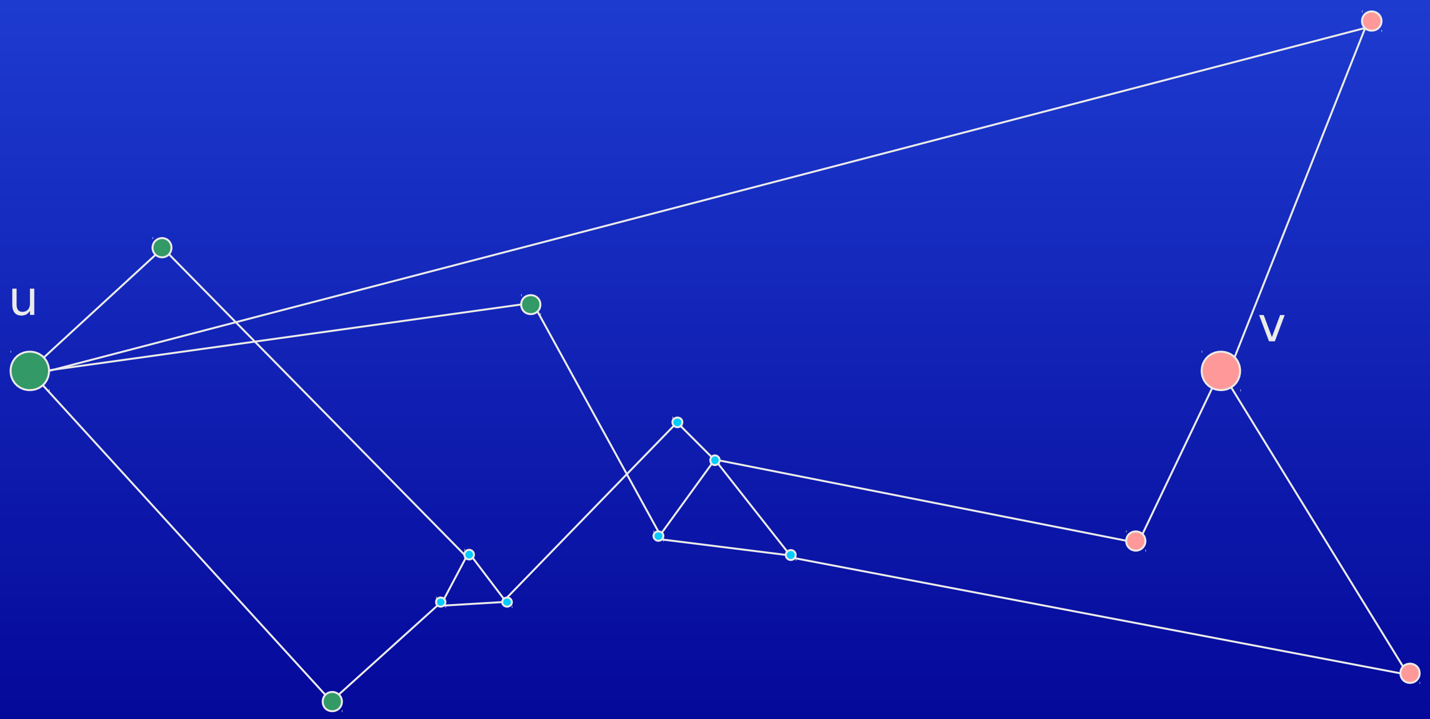
Query: dist(u,v) in $G \setminus \{w_1 \ldots w_k\}$ given $L(u), L(v), L(w_1), \ldots, L(w_k)$

1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size $\tilde{O}(k)$

How does it work?

Query: dist(u,v) in $G \setminus \{w_1 \ldots w_k\}$ given $L(u), L(v), L(w_1), \ldots, L(w_k)$

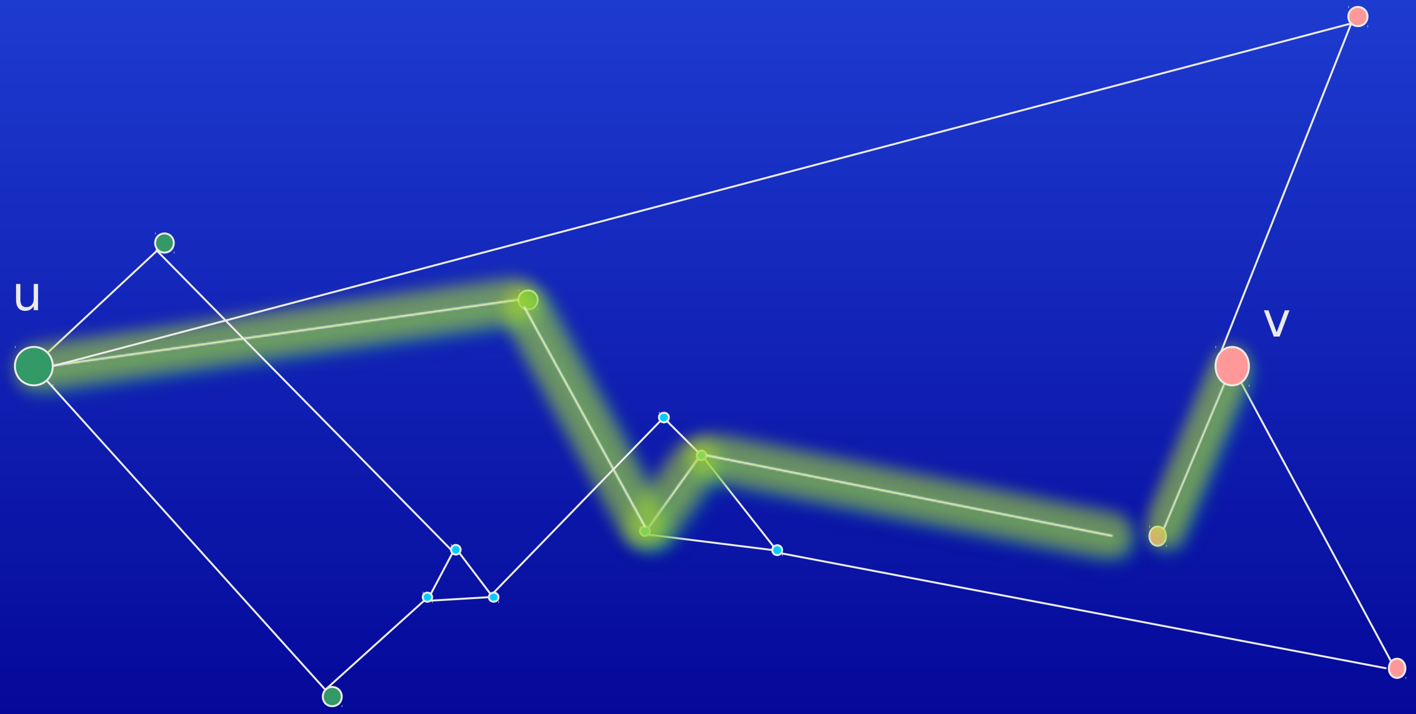
1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size Õ(k)



How does it work?

Query: dist(u,v) in $G \setminus \{w_1 \ldots w_k\}$ given $L(u), L(v), L(w_1), \ldots, L(w_k)$

1. From the labels of the query, construct a sketch graph H
2. Run Dijkstra from u to v in H that has size Õ(k)



Agenda

1. Informative labeling schemes
2. Forbidden-set labeling schemes
3. Challenges

Universal graphs

Induced-universal graph for n-node trees of O(n) size?

Alternatively: design an adjacency labeling scheme for trees with logn+O(1) bit labels?

Best upper bound: $n \cdot 2^{O(\log^* n)}$

FS routing or distance

Design a FS routing scheme for general graphs with short labels?

with similar space/stretch trade-off than fault-free routing, i.e., $\tilde{O}(n^{1/k})$ tables and $O(k)$ stretch

Towards fully-dynamic algorithms

Extend FS labeling scheme to work with some function P(u,v,X,Y,G) between u and v in the graph (G\X)uY.

If query time can be done in Õ(|XuY|) time and if computing all the labels from scratch is near-linear, then it implies lazy sub-linear amortized time fully-dynamic algorithms for maintaining P

That the end

Bye bye Salvador .