# Fast shortest-path queries

Gang : Adrian Kosowski & Laurent Viennot

gang.inria.fr/road

# GANG : algorithms for graphs and networks

Inria – Univ. Paris Diderot – CNRS (10 permanents)

Two themes :

- graph algorithms
- distributed computing
- intersection : network algorithms

Top-down approach :

- from theory to practice,
- upper/lower bounds (what is possible ?)
- what properties allow to do better ?

Question examples :

- How to explore a graph with k agents ?
- How to compactly code distances in a graph ?

Collaborations : mainly Nokia
Example : routing table verification

# Shortest path queries

Problem :

- A graph G is given.
- Answer queries : shortest path from s to t ?

Trivial solution : pre-compute for all $s, t$.

Recent progress [BDG+15], e.g. in road networks ($n = 20$M) :

- Dijkstra : 4s
- Bidirectional Dijkstra : 1s
- Bidirectional $A^*$ : 100ms
- Reach-Pruning, Contraction Hierarchies : 10 ms
- Hub labeling : 10 $\mu$s

# Dijkstra [Dijkstra '59]

**Procedure** Dijkstra $(G, s, t)$

Distance label $d(u) := 0$ if $u = s$, $\infty$.

Radius $r := 0$.

**Repeat**

Pick unscanned $u$ with minimal $d(u)$.

Scan $u$ :

**For** $v \in N_G(u)$ **do** $d(v) := \min\{d(v), d(u) + \ell(uv)\}$

$r := \min_{\text{unscanned } v} d(v)$

**until** $t$ is about to be scanned.

**Return** $d(t)$

# Bidirectional Dijkstra

**Procedure** BidirDijkstra $(G, s, t)$
- Alternate Dijkstra $(G, s, t)$ and Dijkstra $(\overleftarrow{G}, t, s)$.
- Stopping condition?

Estimation $\mu := \infty$ of $d(s, t)$.
When scanning edge uv : $\mu := \min\left\{\mu, d(u) + \ell(uv) + \overleftarrow{d}(v)\right\}$.
Stop if $r + \overleftarrow{r} \geq \mu$.

Exercise : show correctness.

# $A^*$ [Hart, Nilsson, Raphael '68]

Potential function $\pi(\mathsf{u}) \approx \mathsf{d}(\mathsf{u}, \mathsf{t})$.
Dijkstra $(G_\pi, \mathsf{s}, \mathsf{t})$ with $\ell_\pi(\mathsf{uv}) = \ell(\mathsf{uv}) + \pi(\mathsf{v}) - \pi(\mathsf{u})$.
$\mathsf{d}_\pi(\mathsf{s}, \mathsf{t}) = \mathsf{d}(\mathsf{s}, \mathsf{t}) + \pi(\mathsf{t}) - \pi(\mathsf{s})$.
Scan u with $\mathsf{d}(\mathsf{u}) + \pi(\mathsf{u})$ min.

$\pi$ feasible if $\forall \mathsf{uv} \in \mathsf{E}(G)$, $\ell_\pi(\mathsf{uv}) \geq 0$

Bidirectional $A^*$ : ALT [Goldberg, Harelsson '05]
(ALT = $A^*$, Landmarks, Triangle inequality)

- $\pi + \overleftarrow{\pi} = \mathsf{cte}$
- $\pi$ from landmarks

# Reach pruning [Gutman '04] revisited [Goldberg, Kaplan, Werneck '05]

$\text{Reach}(u) = \max_{(s,t) | u \in P_{st}} \min\{d(s,u), d(u,t)\}$

In bidir. Dijkstra, when scanning u :

- Prune v s.t. $\text{Reach}(v) < \min\{d(u) + \ell(uv), \overleftarrow{r}\}$.

Add shortcuts :

- Tie break : fewer links is shorter.

Exercise : how to get shortest path from s to t?

Pre-compute reach upper bounds :

- Eliminate nodes with reach $\leq \delta$.
- Shortcut paths with degree 2 nodes.
- Repeat with larger $\delta$.

Open pb. : graph property ensuring termination.

# Contraction Hierarchies [Sanders, Shultes '05]

Node ordering $\pi : u_1 < \cdots < u_n$

Contract successively $u_i$ :

- add shortcut vw for $v, w \in N(u_i)$ (if needed),
- remove $u_i$ (distances are preserved in remaining graph).

Query : bidir. Dij. in $G^+\uparrow$ and $\overleftarrow{G^+}\uparrow$.

- $G^+$ : graph + shortcuts
- $\uparrow$ : follow uv if $u < v$

Finding $\pi$ :

- small degree + levels (MIS),
- min fill-in (greedy treewidth dec.),
- small balanced separators ($O(n \log n)$ shortcuts if planar).

Exercise : bound shortcut number if any subgraph of $G$ has an $O(n^\varepsilon)$ balanced separator.

Open pb : link between small Reach and small CH ?

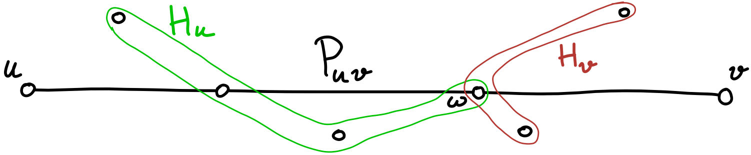
# Distance labeling [Gavoille, Peleg, Pérennes, Raz '04]

**Problem**

Given a graph $G$ assign a label $L_u$ to each node $u$ s.t. for all $s, t$ $d(s,t)$ can be computed from $L_s$ and $L_t$.

**Hub sets**

Given a graph $G$, assign a hub set $H_u \subseteq V(G)$ to each node $u$, s.t. for all $u, v$ there exists $a \in H_u \cap H_v$ with $a \in P_{uv}$.




Distance labels : $L_u = \{(a, d(u,a)) : a \in H_u\}$

Distance query : Dist $(L_s, L_t) = \min_{a \in H_s \cap H_t} d(s,a) + d(a,t)$
in $O(|H_s| + |H_t|)$ time.

# 2-hop labeling [Cohen, Halperin, Kaplan, Zwick '03]

Greedy cover all shortest paths :

- smallest avg. hub size is equivalent to min. cost set cover,
- $O(\log n)$-approximation is possible :
- select $a, S$ s.t. $\frac{\text{nb path cov. if } a \text{ added to all } (H_u)_{u \in S}}{|S|}$ max.

Problem : set cover instance with $n \times 2^n$ sets !

Solution : fix $a$, what is best $S$ ?

$G_a$ graph with edges $uv$ s.t. $P_{uv}$ uncov. and $a \in P_{uv}$.

# Highway dimension [Abraham, Delling, Fiat, Goldberg, Werneck '10-13]

Graph property ensuring small hub sets and efficient ordering for CH.

**Definition**
Highway dimension $h = \max_{u,r} \min_{H \text{ hitting set of } \mathcal{P}_{ur}} |H|$
where $\mathcal{P}_{ur} = \{P \in \mathcal{P}_r \mid \overline{P} \cap B(u,r) \neq \emptyset\}$, $\mathcal{P}_r = \{P \mid \ell(\overline{P}) > \frac{r}{2}\}$,
and $\overline{P}$ is any shortest path extending P by 0 or 1 edge at each extremity.

**Theorem**
Any graph G with highway dimension h and diameter D has hub sets of size $O(h \log D)$ ($O(h \log h \log D)$ for polyn. time).

**Lemma**
For all r, G has an $(h, r)$-sparse hitting set, i.e. a set C s.t. $C \cap P \neq \emptyset$ for all $P \in \mathcal{P}_r$ and $|B(u,r) \cap C| \leq h$ for all $u \in V(G)$.

# Highway dimension [Abraham, Delling, Fiat, Goldberg, Werneck '10-13]

**Theorem**

Any graph $G$ with highway dimension $h$ and diameter $D$ admits a node ordering $\pi$ s.t. $CH_\pi^{opt}$ produces at most $O(nh \log D)$ shortcuts and $CH_\pi + RP$ bidir. Dij. visits $O(h \log D)$ nodes.

**Hierarchical Hub Labeling (HHL) [BGKSW'15]**

A hub labeling is hierarchical if it respects an order $\pi$ such that hubs are more important : $v \in H_u \Rightarrow \pi(u) \leq \pi(v)$ (the graph with edges from nodes to their hubs is a DAG).

**Canonical HHL**

Given an ordering $\pi$, for all $u, v$ add $\max_\pi P_{uv}$ to $H_u$ and $H_v$.

**Proposition**

Canonical HHL for $\pi$ is the minimum HHL that respects $\pi$.

**Exercise** : show that any minimal HHL is canonical.

**Exercise** : use CH to compute a HHL.

# Pruned Labeling [Akiba, Iwata, Yoshida '13]

**Procedure** PrunedLab $(G, \pi)$
  Distance labels $L_u := \emptyset$ for all u.
  **For each** $a \in V(G)$ in decreasing order of $\pi$ **do**
    PrunedDijkstra $(G, a, L)$
    Add $(a, d(a, u))$ to $L_u$ for each scanned node u.

**Procedure** PrunedDijkstra $(G, a, L)$
  Starting from $u = a$, scan u if $d(u) < \mathrm{Dist}(L_a, L_u)$.

**Theorem**
PL computes the canonical HHL associated to $\pi$ in $O(nL^2 + nL\log n + mL)$ time where L is maximum label size.

Open pb : $O(\log n)$ approximation for HHL (find a good $\pi$).
Best is $O(\sqrt{n})$ approx., HHL can be $\sqrt{n}$ larger than HL [BGKSW'15].

Open pb : charac. classes of graphs with $|HHL| = O(|HL|)$.

# HL on massive networks [Delling, Goldberg, Pajor, Werneck '14]

HHL using random sampling to approximate greedy cover (for $\pi$) in combination with pruned labeling (for hub sets).

$O(\log n)$ approximation in theory, smallest hub labelings in practice.

# Skeleton dimension [Kosowski, V. '17]

Graph property ensuring small hub sets and efficient ordering for CH.

The skeleton dimension $k$ of $G$ is the maximum "width" of a "pruned" shortest path tree (see pres.).

**Theorem**
Any graph $G$ with skeleton dimension $k$ and diameter $D$ has hub sets of size $O(k \log\log k \log D)$ (polyn. time constr. w.h.p.).

Open pb : what additional property ensures efficient Reach/CH ?

Open pb : HHL vs HL in grids ?

# Pre-hub labeling [Angelidakis, Makarychev, Oparin '17]

Hub sets $(H_u)_{u \in V(G)}$ for a graph $G$ form a pre-hub labeling if for all $u, v$ pairs, hubs cross on $P_{uv}$ : $\exists u' \in P_{uv} \cap H_u$ and $\exists v' \in P_{uv} \cap H_v$ with $u' \in P_{v'v}$ and $v' \in P_{uu'}$.

**Theorem**

If shortest paths are unique,

- PHL 2-approximate HL (and pol. time constr.),
- PHL can be converted to HL with $O(\log D)$ factor.

**Theorem**

If shortest paths are not unique,

- best polyn. time approx. is $\Omega(\log n)$ (even if $D = O(1)$).

**Theorem**

In trees, HHL 2-approximate HL (and pol. time constr.).

# Public transport

# Transfer Patterns [Bast, Carlsson, Eigenwillig, Geisberger, Harrelson, Raychev, Viger '16]

Pre-compute for each station u :

- lines going through u and position on line,
- transfer pattern DAG : union of $u \to x \to y \to v$ for all v

(and t).

For a query form u to v :

- try direct connection (common line ?),
- otherwise try all possible paths in the DAG of u.

Optimisation : compute global transfer pattern DAG only for a selected set of (common) "hubs".
Prune transfer pattern DAG of u at hubs (local computation).
For a query, use the union of transfer pattern DAGs of u and v.

Experimental results (New York) : precomput. 800h, 1.5Go, transf. pat. 20/station, query time 10ms.

# Public Transit Labeling [Delling, Dibbelt, Pajor, Renato, Werneck '15]

HL on the time expanded graph :

- a node for each event at a station,
- waiting arcs between consecutive nodes of a station,
- connection arcs for each connection.

Query from u to v at t :

- find (by dichotomy) the first event $u'$ at u after t,
- the first event $v'$ s.t. $H_{u'} \cap H_{v'} \neq \emptyset$,
- expand the path through the best hub in $H_{u'} \cap H_{v'}$.

Experimental results (London, earliest arrival – multicriteria) : precomput. 1-49h, 1.3-28Go, hub sets 70-700/event, query time 10-30$\mu$s.

To check : labels per station (more compact) [Wang, Lin, Yang, Xiao, Zhou '15].

# MultiMod challenges

# Robustness of labeling methods to real-time delays?

In case of real-time delays, paths produced by labeling methods are not guaranteed to be optimal.

To do :

- How much stretch compared to optimal?
- How increasing labels to cover more routes reduces stretch?

We need data :

- Evolution of traversal times of road segments within busy hours?
- Real time data from public transport comparable to schedule?

# Similar question : robustness to multi-criteria?

To do :
Compute hub sets for each criterion.
How does the paths covered by their union compare to the set of Pareto optimal routes? How to cover better the Pareto set?

# Challenging multimodal case studies

Efficient methods for fastest bus – bicycle – bus trip?

Efficient methods for fastest bus – bicycle – bus trip?

for closest point from a route?

for closest route from a point?