Fast shortest-path queries

Laurent Viennot

MPRI – Graph Mining (resp. Mauro Sozio)

Shortest path queries

Problem :

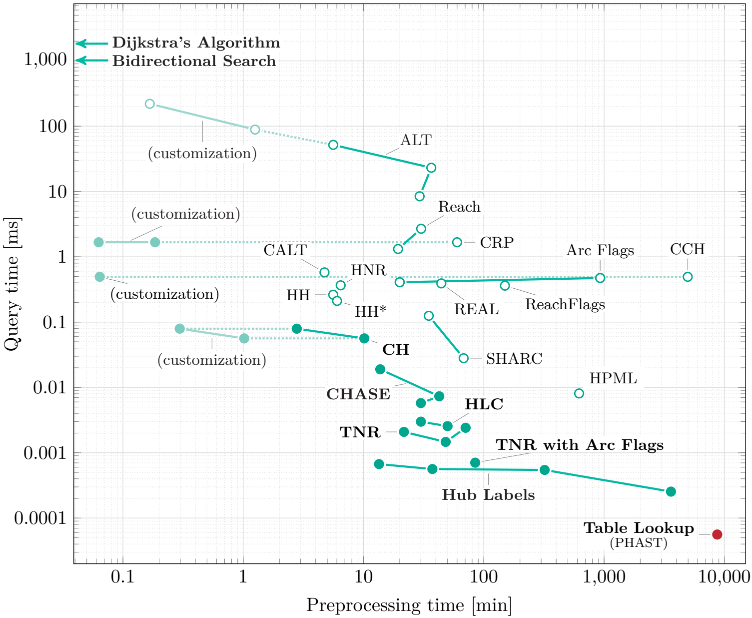
- A graph G is given.
- Answer queries : shortest path from s to t ?

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

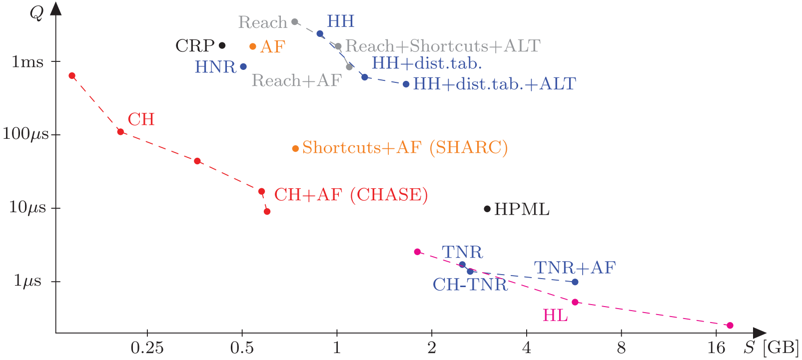
Recent progress [BDG+15], e.g. in road networks ($n = 20$M) :

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

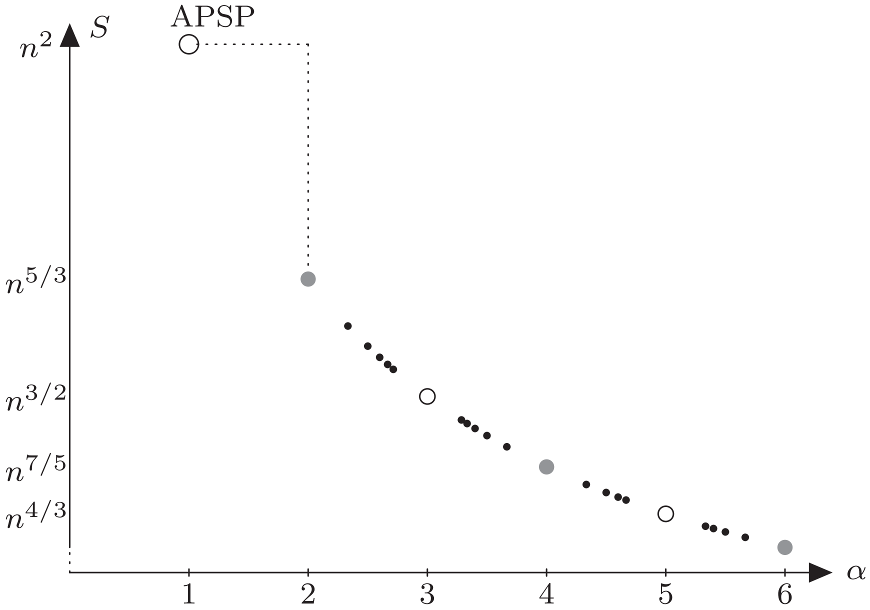
Query time vs preprocessing time (exact) [BDG+15]



Query time vs space (exact)



Space vs stretch (approximate) [Sommer14]



Dijkstra [Dijkstra '59]

Procedure Dijkstra (G, s, t)

- Distance label $d(u) := 0$ if $u = s$, ∞.
- 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.

Exercise : explain 1s vs 4s in road networks.

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

Potential function $\pi(u) \approx d(u,t)$.
Dijkstra (G_π, s, t) with $\ell_\pi(uv) = \ell(uv) - (\pi(u) - \pi(v))$.
$d_\pi(s,t) = d(s,t) - (\pi(s) - \pi(t))$.
Scan u with $d(u) + \pi(u)$ min.

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

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

- $\ell_\pi(uv) = \overleftarrow{\ell}_{\overleftarrow{\pi}}(vu) \iff \pi + \overleftarrow{\pi} = cte$
  (ex : $\pi' = (\pi - \overleftarrow{\pi})/2$ and $\pi' = (\overleftarrow{\pi} - \pi)/2$)
- or different stopping condition
- π from landmarks (better than using coordinates)

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 δ.

Open pb. : graph property ensuring termination.

Contraction Hierarchies [Geisberger, Sanders, Shultes, Delling '05-08]

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 $u <_\pi v$

Finding π :

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

Exercise : bound the number of shortcuts if any subgraph of G has an $O(n^\varepsilon)$ balanced separator and maximum degree Δ.

Open pb : link between small Reach and small CH ?

CH complexity for planar graphs 1/3.

Theorem [Lipton, Tarjan '79] Every planar graph G has a 2/3-balanced separator S_0 of size $O(\sqrt{n})$.

Elimination ordering π : recursively order each connected component and then add nodes in S_0 in any order. This results in a tree of separators of depth $O(\log n)$. (Nested dissection as in [Gilbert, Tarjan '87].)

Rq : all shortcuts occur between a node and an ancestor (possibly the same node).

Corollary : The nodes visited during a pruned Dijkstra from a node s at depth k are all in the separators in the branch of s and query time is $O(\sum_{i=0}^{k} s_i) = O(\sqrt{n})$ (where $s_i = c\sqrt{(2/3)^i n}$ is the maximum size of a separator at depth i).

Lemma : There are $O(n)$ shorcuts with an extremity in S_0.

Rq1 : At most $|S_0|^2 = O(n)$ inside S_0.

Rq2 : Each shortcut with a node u at depth k is the result of the contraction of nodes in the subtree rooted at the separator S containing u.

Def : define the bipartite graph G_k with vertex set $S_0 \cup D_k$ where D_k is the set of separators at depth k and an edge (v, S) for each node $u \in S_0$ and separator $S \in D_k$ such that there exists a shortcut from $u \in S$ to v.

Rq3 : G_k is planar because it can be obtained from G by removing edges inside S_0, removing nodes at depth within 1 and $k-1$ and contracting edges between nodes at depth k or deeper. We thus have $m(G_k) \leq 3n(G_k) - 6$ (use $n + f - m = 2$).

Rq4 : The number of shortcuts with an extremity at depth k is $N_k \leq \sum_{S \in D_k} |S| \deg_{G_k}(S)$. If $D'_k = \{S : \deg_{G_k}(S) > 3\}$, then $N_k \leq \sum_S 3|S| + \sum_{S \in D'_k} s_k(\deg_{G_k}(S) - 3) \leq 3n_k + s_k(m(G'_k) - 3|D'_k|)$ where $G'_k = G_k[D'_k \cup S_0]$ satisfies $m(G'_k) \leq 3n(G'_k)$ implying $N_k \leq 3n_k + 3s_k|S_0|$. The Lemma follows from $\sum_k s_k = O(\sqrt{n})$.

CH complexity for planar graphs 3/3.

Corollary : The number of shortcuts generated by contracting G according to π is $O(n \log n)$.

Rq1 : Similarly to the previous Lemma, for each separator S root of a subtree of n' nodes, there are $O(n')$ shortcuts with an extremity in S.

Rq2 : When summing all subtree sizes, a node is counted $O(\log n)$ times.