Advanced algorithms for the $s - t$ shortest path problem: highway hierarchies

Giovanni Righini

University of Milan

UNIVERSITÀ DEGLI STUDI DI MILANO

References

Main references:

- Highway hierarchies: P. Sanders, D. Schultes, *Highway Hierarchies Hasten Exact Shortest Path Queries*, Annual European Symposium on Algorithms, LNCS 3669 (568-579), 2005.

Highway hierarchies: definitions

In a graph, let the s-rank of a vertex i, $r_s(i)$ the position of vertex i in the list of the vertices permanently labeled by Dijkstra algorithm running from s ($r_s(s) = 0$).
A suitable tie-breaking rule must be defined to ensure that the s-rank of each vertex is unique from each s (canonical shortest paths).

For any given vertex s, the distance of the H-closest vertex from s is denoted by $d_H(s)$: $d_H(s) = dist(s, v)$, where $r_s(v) = H$.

The H-neighbourhood $N_H(s)$ of s is
$N_H(s) = \{i \in \mathcal{N} : dist(s, i) \leq d_H(s)\}$.

On digraphs, symmetrical definitions (distances to t) apply.

Highway hierarchies: definitions

Definition (highway network). For a given value of the parameter H, the highway network $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ of a graph $\mathcal{G}$ is defined as the set of edges $[u, v] \in \mathcal{E}$ that appear in at least one canonical $(s - t)$ shortest path $(s, \ldots, u, v, \ldots, t)$ with the property that $v \notin N_H(s)$ and $u \notin N_H(t)$.

The set $\mathcal{V}_1$ is the maximal subset of $\mathcal{V}$ such that $\mathcal{G}_1$ contains no isolated vertices.

Definition (2-core). The 2-core of a graph is the maximal subgraph with minimum degree two. A graph consists of its 2-core and attached trees, i.e., trees such that their roots belong to the 2-core, but all other nodes do not.

Definition (line). A line in a graph is a path $(u_0, u_1, \ldots, u_k)$, where the inner vertices have degree two.

Highway hierarchies: definitions

Definition (Contracted highway network). From the highway network $\mathcal{G}_1$ of a graph $\mathcal{G}$, the contracted highway network $\mathcal{G}'_1$ of $\mathcal{G}$ is obtained by taking the 2-core of $\mathcal{G}_1$, removing the inner vertices of all lines and replacing each line by a single edge between its endpoints.

Thus, the highway network $\mathcal{G}_1$ consists of the contracted highway network $\mathcal{G}'_1$ and some additional components (trees or lines).

The highway network can be contracted in time $O(m+n)$.

Definition (highway hierarchy). The highway hierarchy of a graph $\mathcal{G}$ is obtained by applying this contraction iteratively.

The algorithm

Pre-processing.
For each vertex $s \in \mathcal{V}$, $d_H(s)$ is computed by growing a shortest path tree from s with Dijkstra algorithm and stopping it as soon as H nodes have received a permanent label.

Construction.
We start with an empty set of highway edges $\mathcal{E}_1$.

For each vertex s, two operations are executed:

- forward construction of a partial shortest path tree B;
- backward evaluation of B.

Forward construction

A Dijkstra search from s is executed.

During the search, a reached vertex is either active or passive.

The source node s is active.

Each vertex that is reached for the first time (*Insert*) and each reached vertex that is updated (*DecreaseKey*) adopts the same activation state from its (tentative) predecessor in the shortest path tree B.

When a vertex p is made permanent using the path $(s, u_1, \ldots, u_k, p)$, then the state of p is set to passive if $|N_H(u_1) \cap N_H(p)| \leq 1$ (i.e. p is "far enough" from u_1).

When no active vertex is left in the priority queue (set of "open" vertices), the growth of B stops.

Backward evaluation: selection of highway edges

All edges $[u, v]$ are added to $\mathcal{E}_1$ if they lie on paths $(s, \ldots, u, v, \ldots, t)$ in B with the property that $v \notin N_H(s)$ and $u \notin N_H(t)$, where t is a leaf of B.

This can be done in time $O(|B|)$ for each source node s.

To speed up the construction, an active vertex v is declared to be a maverick if $dist(s, v) > f\ d_H(s)$, where f is a parameter.

When all active vertices are mavericks, the search from passive vertices is no longer continued.

The pre-processing is accelerated, $\mathcal{E}_1$ becomes a superset of the highway network, queries will be slower, but still exact.

The maverick factor f allows to adjust the trade-off between pre-processing time and query time.

Hierarchy

The highway hierarchy of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of the graphs $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_L$, arranged in $L + 1$ levels.

For each vertex $v \in \mathcal{V}$ and each level ℓ such that $v \in \mathcal{V}_\ell$, there is a copy v_ℓ of vertex v in level ℓ.

In the same way, there are several copies of an edge $[u, v]$ when both u and v belong to more than one common level.

Edges between vertices in the same level are horizontal edges.

Additionally, the hierarchy contains an edge $[v_\ell, v_{\ell+1}]$ for each pair of vertices $v_\ell \in \mathcal{V}_\ell$ and $v_{\ell+1} \in \mathcal{V}_{\ell+1}$ that are copies of the same vertex v in consecutive levels.

These additional edges are called vertical edges and have zero cost.

Hierarchy

For each vertex v and for each level $\ell < L$, the radius $d_H^\ell(v)$ is the distance from v to the H-closest vertex in $\mathcal{G}'_\ell$.

If v does not belong to $\mathcal{G}'_\ell$ then $d_H^\ell(v)$ is set to ∞.

In the last level, $d_H^L(v)$ is set to ∞ for all vertices.

Neighborhoods $N_H^\ell(v)$ are also computed for all vertices and levels: $N_H^\ell(v) = \{v' \in \mathcal{V}'_\ell : dist(v, v') \leq d_H^\ell(v)\}$ is the neighbourhood of v in $\mathcal{G}'_\ell$.

Remark. The neighourhood of a vertex belonging to a component (tree or line) contains all the vertices of the corresponding level. The same holds for to $N_H^L(v)$, for any v.

Query

The multi-level query algorithm is a slight modification of bi-directional Dijkstra algorithm on the hierarchy graph.

The endpoints s and t are s_0 and t_0 at level 0.

Entrance points in level ℓ:

- vertices s_ℓ and t_ℓ;
- any vertex of the core, permanently labeled from a horizontal predecessor out of the core;
- any vertex permanently labeled from a vertical predecessor.

A corresponding entrance point for a permanently labeled vertex v is the last entrance point along the path to v.

Restriction 1. In each level ℓ, no horizontal edge is used that leaves the neighbourhood $N^\ell(v^*)$ of the corresponding entrance point v^*.

Query

Restriction 2. Components (trees and lines) are never entered using a horizontal edge.

An edge $[u, v]$ enters a component if either u belongs to the core and v does not or u belongs to a line and v to an attached tree.
Any edge from an attached tree to a line leaves the attached tree and therefore it does not rank among the edges that enter a component.

Remark. The endpoint(s) of a component do not belong to the component but to the core (or to the line in case of the root of a tree that is attached to a line).

When restriction 1 applies, the search continues on the next level. Horizontal edges that cannot be used in one direction owing to restriction 2, can be used in the opposite direction.

Collapsing the hierarchy in a single level

It is not needed to explicitly represent all levels of the hierarchy.

It is sufficient that at most one copy of each vertex is reached horizontally: the copy with the smallest label and, in case of ties, the one on the lowest level.

Therefore it is enough to store the original graph with some additional pieces of information:

- each edge $[u, v]$ is assigned a maximum level $\lambda(u, v)$, i.e. it belongs to $\mathcal{G}_\ell \ \forall \ell = 0, \ldots, \lambda(u, v)$;
- each vertex v is assigned to at most one component $c(v)$;
- each component belongs to the level ℓ of its inner edges;
- the value $d_H^\ell(v)$ is stored only if $v \in \mathcal{G}'_\ell$.

Stop criterion

The algorithm cannot stop when a vertex v is permanently labelled forward and backward: there is no guarantee that all vertices within a given distance have been already labeled (as in bi-directional Dijkstra algorithm).

Let $\mathcal{E}_s$ and $\mathcal{E}_t$ be the sets of horizontal edges that have been skipped during the search from s and t respectively.

When both search scopes meet, the algorithm can stop as soon as the search from t has finished searching level $\hat{\ell}_s = \max_{e \in \mathcal{E}_s}\{\lambda(e)\}$ and the search from s has finished searching level $\hat{\ell}_t = \max_{e \in \mathcal{E}_t}\{\lambda(e)\}$.

A level ℓ is finished when there are no open vertices in it or below. If the level ℓ is finished, edges e in levels lower than ℓ cannot be used any longer. Hence, when the search from t has finished searching level $\hat{\ell}_s$, it is guaranteed that no edge e in level $\ell \leq \hat{\ell}_s$ would be used even if the algorithm could go on.

Reach and Highway hierarchies

Define c-reach (cardinality reach) of a vertex.

Given v along a shortest $s-t$ path, $P^*(s,t)$, grow equal cardinality balls around s and t until v is included in one of them.
Let $c_{P^*(s,t)}(v)$ the cardinality of the two balls at that point.

$$c(v) = \max_{(s,t):v \in P^*(s,t)} \{c_{P^*(s,t)}(v)\}.$$

For a vertex v and a non-negative integer k, let $\rho(v,k)$ be the radius of the smallest ball centered at v that contains k vertices.

When searching for a shortest $s-t$ path we do not need to scan v if

$$\rho(s, c(v)) < dist(s,v) \wedge \rho(t, c(v)) < dist(v,t).$$

This would require keeping $n-1$ values of ρ for each vertex.

Reach and Highway hierarchies

The partial tree algorithm is used for c-reaches instead of reaches.

Given a threshold H, the algorithm identifies vertices with c-reach below H (local vertices).

Consider a bi-directional search. During the search from s, once the search radius advances past $\rho(s, H)$, one can prune local vertices (and the same backward).

This idea is applied recursively to the graph with low c-reach vertices deleted.

This gives a hierarchy of vertices, in which each vertex needs to store a ρ value for each level of the hierarchy it belongs to.