Highway and VC Dimensions: from Practice to Theory and Back

Andrew V. Goldberg

Microsoft Research – Silicon Valley
http://research.microsoft.com/∼goldberg/

Joint work with
Ittai Abraham, Daniel Delling, Amos Fiat, and Renato Werneck

Theory vs. Practice

Natural Science

- “Science is divided into natural and unnatural.”
  Vladimir Steklov, *mathematician*

Theory vs. Practice

Natural Science

- "Science is divided into natural and unnatural."
  Vladimir Steklov, *mathematician*
- "No Sir, into social and anti-social."
  Sergei Platonov, *historian*

Theory vs. Practice

Natural Science

- “Science is divided into natural and unnatural.”
  Vladimir Steklov, *mathematician*
- “No Sir, into social and anti-social.”
  Sergei Platonov, *historian*

Common Ground: Scientific Method

- Make an observation.
- Form a hypothesis or theory.
- Make a prediction.
- Verify by experiment.

Theory vs. Practice

Natural Science

- “Science is divided into natural and unnatural.”
  Vladimir Steklov, *mathematician*
- “No Sir, into social and anti-social.”
  Sergei Platonov, *historian*

Common Ground: Scientific Method

- Make an observation.
- Form a hypothesis or theory.
- Make a prediction.
- Verify by experiment.

Useful theory.

Scientific Method and Algorithms

- Current technology, practical problem, previous experience.
- Modeling.
- Algorithm design and analysis.
- Experimental evaluation or practical use.

Scientific Method and Algorithms

- Current technology, practical problem, previous experience.
- Modeling.
- Algorithm design and analysis.
- Experimental evaluation or practical use.

Remarks

- Some algorithms are never implemented.
- Some algorithms are never analyzed.
- Scientific method includes both analysis and experimentation.
- Leads to better algorithms.

Scientific Method and Algorithms

- Current technology, practical problem, previous experience.
- Modeling.
- Algorithm design and analysis.
- Experimental evaluation or practical use.

Remarks

- Some algorithms are never implemented.
- Some algorithms are never analyzed.
- Scientific method includes both analysis and experimentation.
- Leads to better algorithms.

Mathematics of algorithms vs. algorithm engineering vs. algorithm science: algorithm research via the sceintific method.

Outline

Shortest Paths: Recent Developments

Continent-sized road networks have 10s of millions intersections.

Dijkstra's algorithm: ≈ 5 s

Shortest Paths: Recent Developments

Continent-sized road networks have 10s of millions intersections.

Dijkstra's algorithm: $\approx$ 5 s

Recent work

- Arc flags [Lauther 04, Köhler et al. 06, Bauer & Delling 08].
- A^* with landmarks [Goldberg & Harrelson 05].
- Reach [Gutman 04, Goldberg et al. 06].
- Highway hierarchies [Sanders & Schultes 05].
- Contraction hierarchies [Geisberger et al. 08].
- Transit nodes [Bast et al. 06].
- DIMACS Shortest Paths Implementation Challenge (2005–2006).

Shortest Paths: Recent Developments

Continent-sized road networks have 10s of millions intersections.

Dijkstra's algorithm: ≈ 5 s

Recent work

- Arc flags [Lauther 04, Köhler et al. 06, Bauer & Delling 08].
- A^* with landmarks [Goldberg & Harrelson 05].
- Reach [Gutman 04, Goldberg et al. 06].
- Highway hierarchies [Sanders & Schultes 05].
- Contraction hierarchies [Geisberger et al. 08].
- Transit nodes [Bast et al. 06].
- DIMACS Shortest Paths Implementation Challenge (2005–2006).

Greatly improved performance: < 1 ms, ≈ 0.1 s on a mobile device.
Only a few hundred intersections searched.

Definitions and Model

Input

- Graph $G = (V, E)$ (intersections, road segments), $|V| = n$, $|E| = m$.
- Weight function ℓ (length, transit time, fuel consumption, ...).
- Static problem, G and ℓ incorporate all modeling information.

Definitions and Model

Input

- Graph $G = (V, E)$ (intersections, road segments), $|V| = n$, $|E| = m$.
- Weight function ℓ (length, transit time, fuel consumption, ...).
- Static problem, G and ℓ incorporate all modeling information.

Query (multiple times for the same input network)

- Given origin s and destination t, find optimal path from s to t.
- Exact algorithms help modeling and debugging.

Definitions and Model

Input

- Graph $G = (V, E)$ (intersections, road segments), $|V| = n$, $|E| = m$.
- Weight function ℓ (length, transit time, fuel consumption, ...).
- Static problem, G and ℓ incorporate all modeling information.

Query (multiple times for the same input network)

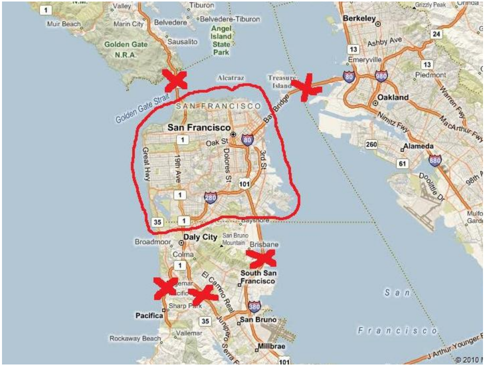
- Given origin s and destination t, find optimal path from s to t.
- Exact algorithms help modeling and debugging.

Algorithms with preprocessing

- Two phases: practical preprocessing and real-time queries.
- Preprocessing output not much bigger than the input.
- Preprocessing may use more resources than queries.

Transit Node (TN) Algorithm

[Bast et al. 06]



For a region, there is a small set of nodes such that all sufficiently long shortest paths out of the region pass a node in the set.

TN Preprocessing

Basic concepts

- Divide a map into regions (a few thousand).
- For each region, optimal paths to far away places pass through one of a small number of access nodes (≈ 10 on the average).
- The union of access nodes is the set of transit nodes ($\approx 10\,000$).

TN Preprocessing

Basic concepts

- Divide a map into regions (a few thousand).
- For each region, optimal paths to far away places pass through one of a small number of access nodes (≈ 10 on the average).
- The union of access nodes is the set of transit nodes ($\approx 10\,000$).

Empirical observation: small number of access/transit nodes.

TN Preprocessing

Basic concepts

- Divide a map into regions (a few thousand).
- For each region, optimal paths to far away places pass through one of a small number of access nodes (≈ 10 on the average).
- The union of access nodes is the set of transit nodes ($\approx 10\,000$).

Empirical observation: small number of access/transit nodes.

Preprocessing Algorithm

- Find access nodes for every region.
- Connect each vertex to its access nodes.
- Compute all pairs of shortest paths between transit nodes.

TN Query

Long-range query algorithm

- The shortest path has the form s – access(s) – access(t) – t

TN Query

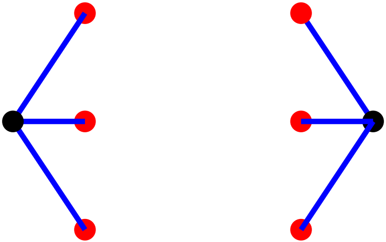
Long-range query algorithm

- The shortest path has the form s – access(s) – access(t) – t

TN Query

Long-range query algorithm

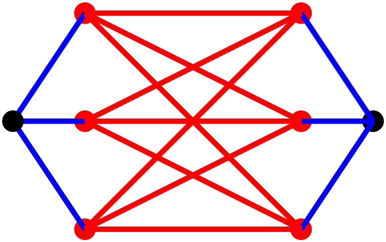
- The shortest path has the form s – access(s) – access(t) – t



TN Query

Long-range query algorithm

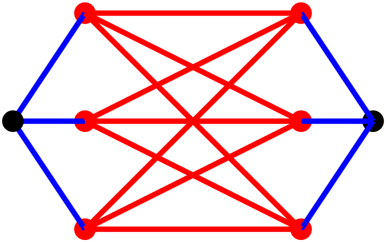
- The shortest path has the form s – access(s) – access(t) – t



TN Query

Long-range query algorithm

- The shortest path has the form s – access(s) – access(t) – t
- Table look-up for the (access(s), access(t)) node pairs.



Remarks

- Very fast: 10×10 table look-ups per long-range query.
- Local queries: another method or hierarchical approach.

Theoretical Results

Practice

- Intuitive and practical algorithms, but...
- Why do they work well on road networks?
- What is a road network (formally)?

Theoretical Results

Practice

- Intuitive and practical algorithms, but...
- Why do they work well on road networks?
- What is a road network (formally)?

Theory [Abraham, Fiat, Goldberg & Werneck '10]

- Define highway dimension (HD).
- Good time bounds for transit nodes, highway hierarchies, and reach algorithms assuming HD is small.
- Analysis highlights algorithm similarities.

Definitions and Theoretical Results

Definitions and assumptions

- Constant maximum degree.
- $B_{v,r}$ denotes the set of vertices within distance r from v.
- $\ell(P)$ denotes the length of P.
- Assume shortest paths are unique.
- h denotes highway dimension.
- Network diameter D.

Definitions and Theoretical Results

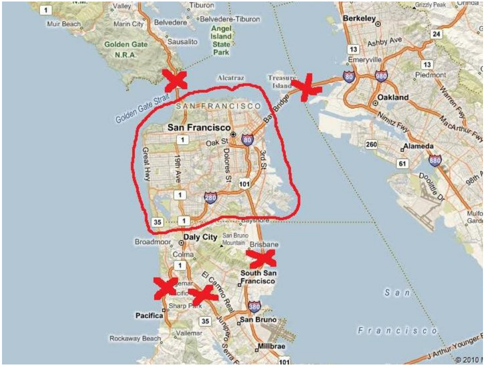
Definitions and assumptions

- Constant maximum degree.
- $B_{v,r}$ denotes the set of vertices within distance r from v.
- $\ell(P)$ denotes the length of P.
- Assume shortest paths are unique.
- h denotes highway dimension.
- Network diameter D.

Theoretical Results

- Polynomial-time preprocessing.
- Query time polynomial in h and $\log D$ ("polylog").
- Space overhead factor polynomial in h and $\log D$.

Highway Dimension Motivation



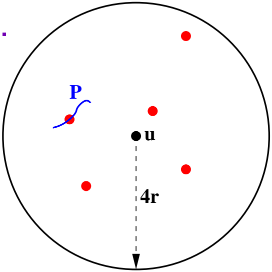
For a region, there is a small set of nodes such that all sufficiently long shortest paths out of the region pass a node in the set.

Highway Dimension Definition

Locally, a small set of vertices hits all long SPs.

Highway Dimension Definition

Locally, a small set of vertices hits all long SPs.



Highway dimension (HD) h

$\forall \quad r \in \Re, \forall u \in V, \exists S \subseteq B_{u,4r}, |S| \leq h,$ such that

$\forall \quad v, w \in B_{u,4r},$

if P is an SP with $\ell(P(v,w)) > r$ and $P(v,w) \subseteq B_{u,4r}$,

then $P(v,w) \cap S \neq \emptyset$.

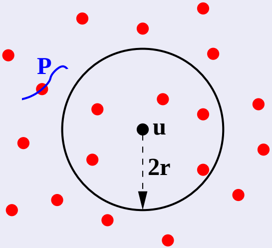
Shortest Path Covers

All SPs in a range can be hit by a sparse set.

(r, k) Shortest path cover ((r, k)-SPC):

A set C such that

$$\forall \quad \text{SP } P: \; r < \ell(P) \leq 2r \Rightarrow$$
$$P \cap C \neq \emptyset \quad \text{and}$$
$$\forall u \in V, \; |C \cap B_{u,2r}| \leq k$$



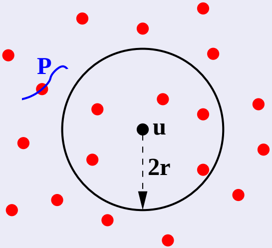
Shortest Path Covers

All SPs in a range can be hit by a sparse set.

(r,k) Shortest path cover ((r,k)-SPC):

A set C such that

$$\forall \quad \text{SP } P:\ r < \ell(P) \le 2r \Rightarrow$$
$$P \cap C \neq \emptyset \quad \text{and}$$
$$\forall u \in V,\ |C \cap B_{u,2r}| \le k$$



Constants 4 (HD definition) and 2 (SPC definition) are related.

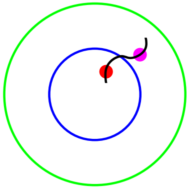
HD vs. SPC

Theorem

If G has HD h, then $\forall\ r\ \exists$ an (r, h)-SPC.

Proof:

- Show S^*, the smallest set hitting all SPs $P:\ r < \ell(P) \leq 2r$, is an (r, h)-SPC.
- Suppose $|S^* \cap B_{v,2r}| > h$.
- Consider $B_{v,4r}$, it contains a set H with $|H| \leq h$ that hits all SPs $P : \ell(P) > r$.
- H hits all SPs $P : r < \ell(P) \leq 2r$ hit by $S^* \cap B_{v,2r}$.
- Replacing $S^* \cap B_{v,2r}$ by H gives a smaller set S^*.



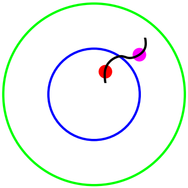
HD vs. SPC

Theorem

If G has HD h, then $\forall\ r\ \exists$ an (r,h)-SPC.

Proof:

- Show S^*, the smallest set hitting all SPs $P:\ r < \ell(P) \leq 2r$, is an (r,h)-SPC.
- Suppose $|S^* \cap B_{v,2r}| > h$.
- Consider $B_{v,4r}$, it contains a set H with $|H| \leq h$ that hits all SPs $P : \ell(P) > r$.
- H hits all SPs $P : r < \ell(P) \leq 2r$ hit by $S^* \cap B_{v,2r}$.
- Replacing $S^* \cap B_{v,2r}$ by H gives a smaller set S^*.



Finding S^* is NP-hard. Efficient construction?

Computing Approximate SPCs

- Greedy approximation for S^* gives an $O(\log n)$ factor approximation.
- Approximation independent of n?

Computing Approximate SPCs

- Greedy approximation for S^* gives an $O(\log n)$ factor approximation.
- Approximation independent of n?

Theorem

Suppose we have a poly-time, $(c \log h)$ approximation algorithm for hitting set. If G has HD h, then for any r we can construct, in polynomial time, an $(r, O(h \log h))$-SPC.

Proof: Similar to the previous proof. Maintain a hitting set S. If for some v, $|S \cap B_{v,2r}| > c \log h$, compute a hitting set for the SPs in $B_{v,4r}$ of size at most $c \log h$ and get a smaller hitting set S.

Computing Approximate SPCs

- Greedy approximation for S^* gives an $O(\log n)$ factor approximation.
- Approximation independent of n?

Theorem

Suppose we have a poly-time, $(c \log h)$ approximation algorithm for hitting set. If G has HD h, then for any r we can construct, in polynomial time, an $(r, O(h \log h))$-SPC.

Proof: Similar to the previous proof. Maintain a hitting set S. If for some v, $|S \cap B_{v,2r}| > c \log h$, compute a hitting set for the SPs in $B_{v,4r}$ of size at most $c \log h$ and get a smaller hitting set S.

Connection to VC-dimension [Vapnik & Chervonenkis 71]

[Brönnimann & Goodrich 95]: $O(hd \log(hd))$ hitting sets for set systems of VC-dimension d.

VC-Dimension

- Base set X, collection of subsets $\mathcal{R}$, set system $(X, \mathcal{R})$.
- For $Y \subseteq X$, $Y_{|\mathcal{R}} = (Y, \{Z \cap Y | Z \in \mathcal{R}\})$.
- $\mathcal{R}$ shatters Y if $Y_{|\mathcal{R}} = 2^Y$.
- $(X, \mathcal{R})$ has VC-dimension d if d is the smallest integer such that no $d+1$ subset of X can be shattered.
- A hitting set intersects all sets in $\mathcal{R}$.

VC-Dimension

- Base set X, collection of subsets $\mathcal{R}$, set system $(X, \mathcal{R})$.
- For $Y \subseteq X$, $Y_{|\mathcal{R}} = (Y, \{Z \cap Y | Z \in \mathcal{R}\})$.
- $\mathcal{R}$ shatters Y if $Y_{|\mathcal{R}} = 2^Y$.
- $(X, \mathcal{R})$ has VC-dimension d if d is the smallest integer such that no $d+1$ subset of X can be shattered.
- A hitting set intersects all sets in $\mathcal{R}$.

Example (points and half-planes)

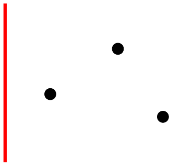
X is a plane, $\mathcal{R}$ is the set of all half-planes, VC-dimension is 3.

VC-Dimension

- Base set X, collection of subsets $\mathcal{R}$, set system $(X, \mathcal{R})$.
- For $Y \subseteq X$, $Y_{|\mathcal{R}} = (Y, \{Z \cap Y | Z \in \mathcal{R}\})$.
- $\mathcal{R}$ shatters Y if $Y_{|\mathcal{R}} = 2^Y$.
- $(X, \mathcal{R})$ has VC-dimension d if d is the smallest integer such that no $d+1$ subset of X can be shattered.
- A hitting set intersects all sets in $\mathcal{R}$.

Example (points and half-planes)

X is a plane, $\mathcal{R}$ is the set of all half-planes, VC-dimension is 3.

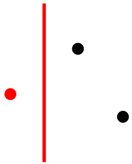


VC-Dimension

- Base set X, collection of subsets $\mathcal{R}$, set system $(X, \mathcal{R})$.
- For $Y \subseteq X$, $Y_{|\mathcal{R}} = (Y, \{Z \cap Y | Z \in \mathcal{R}\})$.
- $\mathcal{R}$ shatters Y if $Y_{|\mathcal{R}} = 2^Y$.
- $(X, \mathcal{R})$ has VC-dimension d if d is the smallest integer such that no $d + 1$ subset of X can be shattered.
- A hitting set intersects all sets in $\mathcal{R}$.

Example (points and half-planes)

X is a plane, $\mathcal{R}$ is the set of all half-planes, VC-dimension is 3.

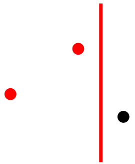


VC-Dimension

- Base set X, collection of subsets $\mathcal{R}$, set system $(X, \mathcal{R})$.
- For $Y \subseteq X$, $Y_{|\mathcal{R}} = (Y, \{Z \cap Y | Z \in \mathcal{R}\})$.
- $\mathcal{R}$ shatters Y if $Y_{|\mathcal{R}} = 2^Y$.
- $(X, \mathcal{R})$ has VC-dimension d if d is the smallest integer such that no $d + 1$ subset of X can be shattered.
- A hitting set intersects all sets in $\mathcal{R}$.

Example (points and half-planes)

X is a plane, $\mathcal{R}$ is the set of all half-planes, VC-dimension is 3.

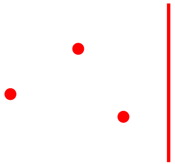


VC-Dimension

- Base set X, collection of subsets $\mathcal{R}$, set system $(X, \mathcal{R})$.
- For $Y \subseteq X$, $Y_{|\mathcal{R}} = (Y, \{Z \cap Y | Z \in \mathcal{R}\})$.
- $\mathcal{R}$ shatters Y if $Y_{|\mathcal{R}} = 2^Y$.
- $(X, \mathcal{R})$ has VC-dimension d if d is the smallest integer such that no $d+1$ subset of X can be shattered.
- A hitting set intersects all sets in $\mathcal{R}$.

Example (points and half-planes)

X is a plane, $\mathcal{R}$ is the set of all half-planes, VC-dimension is 3.

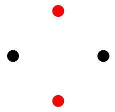


VC-Dimension

- Base set X, collection of subsets $\mathcal{R}$, set system $(X, \mathcal{R})$.
- For $Y \subseteq X$, $Y_{|\mathcal{R}} = (Y, \{Z \cap Y | Z \in \mathcal{R}\})$.
- $\mathcal{R}$ shatters Y if $Y_{|\mathcal{R}} = 2^Y$.
- $(X, \mathcal{R})$ has VC-dimension d if d is the smallest integer such that no $d + 1$ subset of X can be shattered.
- A hitting set intersects all sets in $\mathcal{R}$.

Example (points and half-planes)

X is a plane, $\mathcal{R}$ is the set of all half-planes, VC-dimension is 3.



VC-Dimension and Shortest Paths

- X is the set of vertices.
- $\mathcal{R}$ contains the sets of vertices on SPs $P: \ r < \ell(P) \leq 2r$.
- VC-dimension of $(X, \mathcal{R})$ is at most two.

VC-Dimension and Shortest Paths

- X is the set of vertices.
- $\mathcal{R}$ contains the sets of vertices on SPs $P: \ r < \ell(P) \le 2r$.
- VC-dimension of $(X, \mathcal{R})$ is at most two.

Theorem

[Brönnimann & Goodrich 95]: If h is the optimal hitting set size and d is VC-dimension, then we can find an $O(hd \log(hd))$ hitting set in polynomial time.

Corollary

For an HD h graph, we can efficiently compute an $O(h \log h)$-size hitting set for SPs $P: \ r < \ell(P) \le 2r$.

BG Algorithm Outline

Algorithm has a learning flavor

1. Start with all vertices having weight one.
2. Pick a random weighted set S of size $(ch \log h)$.
3. If S is a hitting set halt.
4. Find an SP P that is not covered, double vertex weights on P.
5. Goto 2.

BG Algorithm Outline

Algorithm has a learning flavor

1. Start with all vertices having weight one.
2. Pick a random weighted set S of size $(ch \log h)$.
3. If S is a hitting set halt.
4. Find an SP P that is not covered, double vertex weights on P.
5. Goto 2.

Remarks

- The algorithm can be derandomized.
- Not the first algorithm for SPC one would think of.
- Currently, our best SPC algorithm uses these ideas.

Work in Progress

Implications and refinement of theory

- Practical SPC algorithms for big networks.
- Computing HD of real maps.
- Alternative HD definitions.
- Improved algorithms explicitly based on SPCs.
- Other practical applications of SPCs.
- Other theoretical applications of HD (e.g., Steiner Tree construction).

Scientific method

From practice to theory to practice.

Thank You!

SPA (Shortest Path Algorithms) project page

http://research.microsoft.com/en-us/projects/SPA/

Questions?

1. Recent Shortest Path Algorithms
2. Transit Node Algorithm
3. Highway Dimension and Shortest Path Covers (SPCs)
4. Computing SPCs
5. VC-Dimension
6. Work in Progress