Exact and Approximate Shortest-Path Queries for Planar Graphs

Christian Sommer
csom@mit.edu  www.sommer.jp

Joint work with Shay Mozes (exact q.) and Philip N. Klein and Ken-ichi Kawarabayashi (approximate)

15 February 2011, NII Shonan Workshop
Motivation
27. Turn left at 6th Ave NE

28. Turn right at NE Northlake Way

29. Kayak across the Pacific Ocean

30. Continue straight

31. Turn left at Kuilima Dr
Shortest-Path Queries / Distance Oracles

- Preprocess a graph $G$ with $n$ nodes and $m$ edges ...
- ... to create a Data Structure, using which ...
- ... we can efficiently answer Distance Queries.
  - $d(u, v)$
Shortest-Path Queries / Distance Oracles

- Preprocess a graph $G$ with $n$ nodes and $m$ edges ...

- ... to create a Data Structure, using which ...

- ... we can efficiently answer Approximate Distance Queries.

  - $\tilde{d}(u, v)$
Approximate Distance Oracles — Stretch

- Distance between \( u \) and \( v \) in graph \( G \): \( d_G(u, v) \)
- Oracle Result \( \tilde{d}(u, v) \) satisfies

\[
d_G(u, v) \leq \tilde{d}(u, v) \leq \alpha \cdot d_G(u, v).
\]

- Multiplicative Stretch \( \alpha \)
Related Work

Practical

- Focus on Transportation Networks

Theoretical

- General, undirected graphs
- Restricted classes (planar, bounded tree-width, minor-closed,...)
Shortest-Path Queries in Transportation Networks

- Main focus, large body of research since 60’s/70’s
- Big progress around 2006 (DIMACS Implementation Challenge)
  - Preprocessing: tens of minutes for road map of the US/EU
  - Query time: $\approx 10^6$ times faster than Dijkstra’s algorithm

- Ideas
  - Geometry, coordinates, A* search [SV86]
  - Goal-directed search (A* for graphs) [GH05]
  - Hierarchical structures [SS05, BFSS07, BD08, BDS+08]
- Heuristics that work very well for road networks (separators)
- Visit only edges of shortest path
Space vs. Query Time Tradeoffs: Experimental Results

Diagram showing the tradeoffs between space and query time with various algorithms and data structures. The graph is plotted with axes representing space and query time, with points labeled as follows:

- HH (basic)
- edge flags
- HNR
- CH
- SHARC
- reach + shortc. + A*
- HH + dist. tab.
- TNR
- HL

The points are connected with lines to illustrate the tradeoff trends.
Related Work

Practical

- Focus on Transportation Networks

Theoretical

- General, undirected graphs
- Restricted graph classes (planar, bounded tree-width, minor-closed,...)
Planar Separators, Graph $G = (V, E)$

Partition $V$ into $V_1, V_2, S$ such that $|V_1|, |V_2| \leq \frac{n}{2}$, no edge between $V_1, V_2$, and

- Lipton and Tarjan [LT80], Miller [Mil86]
  - s.t. $|S| = \mathcal{O}(\sqrt{n})$
  - Shortest paths may cross $S$ up to $\mathcal{O}(\sqrt{n})$ times

- Thorup [Tho04]
  - s.t. $S$ consists of 3 shortest paths
  - can be extended to minor-closed families [AG06]

- Dieng and Gavoille [DG09]
  - s.t. $S$ consists of $\mathcal{O}(1)$ shortest paths of length “tree-length”
$r$–divisions [Fre87].

separate recursively (e.g. using [Mil86]) into

- $O(n/r)$ regions
- region size $O(r)$
- region boundary $O(\sqrt{r}) \sim$ total boundary $O(n/\sqrt{r})$
Outline

- Introduction
- Planar Separators
- Approximate Distance Oracles
- Exact Distance Oracles
- Conclusion
$(1 + \epsilon)$–Approximate Shortest-Path Queries; Planar $G$

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>Space</th>
<th>Query</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(n\epsilon^{-2} \lg^4 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg^2 n)$</td>
<td>$O(\epsilon^{-1} + \lg \lg n)$</td>
<td>[Tho04, Thm.3.16]</td>
</tr>
<tr>
<td>$O(n\epsilon^{-1} \lg^3 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg^2 n)$</td>
<td>$O(\epsilon^{-1} + \lg \lg \lg n)$</td>
<td>[Tho04, Prop.3.14]</td>
</tr>
<tr>
<td>$O(n(\epsilon^{-1} + \lg n) \lg^2 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg^2 n)$</td>
<td>$O(\epsilon^{-1} + \lg n \lg \lg n)$</td>
<td>[Kle05, Sec.7]</td>
</tr>
<tr>
<td>$O(n\epsilon^{-2} \lg^4 n)$</td>
<td>$O(n)$</td>
<td>$O(\epsilon^{-2} \lg^3 n)$</td>
<td>NEW</td>
</tr>
<tr>
<td>$O(n\epsilon^{-2} \lg^3 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg n)$</td>
<td>$O(\epsilon^{-1})$</td>
<td>[Tho04, Thm.3.19]</td>
</tr>
<tr>
<td>$O(n\epsilon^{-1} \lg^2 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg n)$</td>
<td>$O(\epsilon^{-1} \lg n)$</td>
<td>[Tho04, Implicit]</td>
</tr>
<tr>
<td>$O(n \lg^2 n)$</td>
<td>$O(n)$</td>
<td>$O(\epsilon^{-2} \lg^2 n)$</td>
<td>NEW</td>
</tr>
</tbody>
</table>

Assumption for this table: largest integer weight $N = O(poly(n))$
(complexity of oracles for planar digraphs depends on $N$)
(1 + $\epsilon$)–Approximate Shortest-Path Queries; Planar $G$

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>Space</th>
<th>Query</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(n \epsilon^{-2} \lg^4 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg^2 n)$</td>
<td>$O(\epsilon^{-1} + \lg \lg n)$</td>
<td>[Tho04, Thm.3.16]</td>
</tr>
<tr>
<td>$O(n \epsilon^{-1} \lg^3 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg^2 n)$</td>
<td>$O(\epsilon^{-1} + \lg n \lg \lg n)$</td>
<td>[Tho04, Prop.3.14]</td>
</tr>
<tr>
<td>$O(n(\epsilon^{-1} + \lg n) \lg^2 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg^2 n)$</td>
<td>$O(\epsilon^{-1} + \lg n \lg \lg n)$</td>
<td>[Kle05, Sec.7]</td>
</tr>
<tr>
<td>$O(n \epsilon^{-2} \lg^4 n)$</td>
<td>$O(n)$</td>
<td>$O(\epsilon^{-2} \lg^3 n)$</td>
<td>NEW</td>
</tr>
<tr>
<td>$O(n \epsilon^{-2} \lg^3 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg n)$</td>
<td>$O(\epsilon^{-1})$</td>
<td>[Tho04, Thm.3.19]</td>
</tr>
<tr>
<td>$O(n \epsilon^{-1} \lg^2 n)$</td>
<td>$O(n \cdot \epsilon^{-1} \lg n)$</td>
<td>$O(\epsilon^{-1} \lg n)$</td>
<td>[Tho04, Implicit]</td>
</tr>
<tr>
<td>$O(n \lg^2 n)$</td>
<td>$O(n)$</td>
<td>$O(\epsilon^{-2} \lg^2 n)$</td>
<td>NEW</td>
</tr>
</tbody>
</table>

Assumption for this table: largest integer weight $N = O(poly(n))$
(complexity of oracles for planar digraphs depends on $N$)
Main Techniques of Thorup’s Distance Oracle

Partition $V$ into $V_1, V_2, S$ such that $|V_1|, |V_2| \leq \frac{n}{2}$ and

- s.t. $S$ consists of 3 shortest paths $Q$
- $\leadsto$ shortest paths cannot cross many times

- Representation of paths that intersect $Q$
shortest path $Q$
shortest path $Q$
$O(1/\epsilon)$ connections $q$

s.t. $d(v, q) + d(q, q') \leq (1 + \epsilon) d(v, q')$
Space Consumption of Thorup’s Distance Oracle

- Recursive partition using 3 shortest paths $Q$ per level
  - $O(\log n)$ shortest-path separators per node

- Representation of paths that intersect $Q$
  - store $O(1/\epsilon)$ connections

- Total storage: $O(\epsilon^{-1} \log n)$ connections per node
Linear-Space Distance Oracle: Main Idea

- store connections for few nodes (landmarks) ⇝ boundary of $r$–division!

- at query time, search landmark then use [Tho04]
\((1 + \epsilon)\)–Approximate Shortest-Path Queries; Planar \(G\)

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>Space</th>
<th>Query</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>(O(n\epsilon^{-2} \lg^4 n))</td>
<td>(O(n \cdot \epsilon^{-1} \lg^2 n))</td>
<td>(O(\epsilon^{-1} + \lg \lg n))</td>
<td>[Tho04, Thm.3.16]</td>
</tr>
<tr>
<td>(O(n\epsilon^{-1} \lg^3 n))</td>
<td>(O(n \cdot \epsilon^{-1} \lg^2 n))</td>
<td>(O(\epsilon^{-1} + \lg n \lg \lg n))</td>
<td>[Tho04, Prop.3.14]</td>
</tr>
<tr>
<td>(O(n(\epsilon^{-1} + \lg n) \lg^2 n))</td>
<td>(O(n \cdot \epsilon^{-1} \lg^2 n))</td>
<td>(O(\epsilon^{-1} + \lg n \lg \lg n))</td>
<td>[Kle05, Sec.7]</td>
</tr>
<tr>
<td>(O(n\epsilon^{-2} \lg^4 n))</td>
<td>(O(n))</td>
<td>(O(\epsilon^{-2} \lg^3 n))</td>
<td>NEW</td>
</tr>
</tbody>
</table>

Assumption for this table: largest integer weight \(N = O(poly(n))\) (complexity of oracles for planar digraphs depends on \(N\))
Outline

- Introduction
- Planar Separators
- Approximate Distance Oracles
- Exact Distance Oracles
- Conclusion
$r$–divisions [Fre87].

separate recursively (e.g. using [Mil86]) into

- $O(n/r)$ regions
- region size $O(r)$
- region boundary $O(\sqrt{r}) \leadsto$ total boundary $O(n/\sqrt{r})$
$r$-divisions [Fre87].

separate recursively (e.g. using [Mil86]) into

- $O(n/r)$ regions
- region size $O(r)$
- region boundary $O(\sqrt{r}) \rightsquigarrow$ total boundary $O(n/\sqrt{r})$

**Idea:** store connections among boundary nodes

- space $O(n^2/r)$
- query time $O(r)$ (explore region, compute min boundary pair)
Space vs. Query Time for Exact Shortest Paths

\[ \frac{\log Q}{\log n}, \quad \frac{\log S}{\log n} \]

[Dji96] [ACC⁺96]
$r$–divisions [Fre87].
How To Use $r$–divisions [Dji96]
Hot To Use $r$–divisions [Dji96]
Space vs. Query Time for Exact Shortest Paths

\[ \frac{\log Q}{\log n} \quad \text{vs.} \quad \frac{\log S}{\log n} \]

[Dji96]

[ACC+96]

\[ \frac{1}{4} \quad \frac{1}{3} \quad \frac{1}{2} \quad 1 \]

\[ 2 \quad 3/2 \quad 4/3 \]
Planarity-Exploiting Search at Query Time [Dji96]
Space vs. Query Time for Exact Shortest Paths

\[ \frac{\log Q}{\log n} \quad \frac{\log S}{\log n} \]

[Dji96]

[ACC+96]

[CX00, Cab06]
Hot To Use $r$–divisions [Cab06]
How To Use $r$–divisions [Cab06]
Efficient Representation of Connection to Boundary
Klein’s MSSP data structure [Kle05]

preprocess $G, f$

$O(n \lg n)$ time & space
Klein’s MSSP data structure [Kle05]

query \(d(q, v)\)

\(O(\lg n)\) time
Efficient Representation of Connection to Boundary
Space vs. Query Time for Exact Shortest Paths

\[ \log S / \log n \]

\[ \log Q / \log n \]

\[ [Dji96] \]

\[ [ACC^+96] \]

\[ [FR06] \]

\[ [CX00, Cab06] \]
Space vs. Query Time for Exact Shortest Paths

\[ \frac{\log Q}{\log n} \quad \frac{\log S}{\log n} \]

[Dji96] [ACC+96] [FR06] [Dji96] [CX00, Cab06]
Space vs. Query Time for Exact Shortest Paths

\[
\frac{\lg Q}{\lg n} \quad \frac{\lg S}{\lg n}
\]

[FR06] [Dji96] [ACC\textsuperscript{+}96] [Dji96] [CX00, Cab06] NEW
New Cycle-MSSP data structure
New Cycle-MSSP data structure

preprocess $G, C$

$\tilde{O}(n)$ time

$O(n \log \log n)$ space

$|C| = O(\sqrt{n})$
New Batched Cycle-MSSP data structure

query $d(C, v)$
$\tilde{O}(|C|)$ time
Space vs. Query Time for Exact Shortest Paths

\[ \frac{\log Q}{\log n} \quad \frac{\log S}{\log n} \]

- [Dji96]
- [ACC+96]
- [FR06]
- [Dji96]
- [CX00, Cab06]

NEW
Space vs. Query Time Tradeoffs: Experimental Results

![Graph showing space vs. query time tradeoffs with various algorithm representations such as edge flags, HH (basic), HH+dist. tab., reach+shortc.+A*, SHARC, TNR, and HL.]
Hilger et al. [HKMS09, Section 6]

In all cases, the search space of our arc-flag method is never larger than ten times the actual number of nodes on the shortest paths.
Can prove that

In all cases, the search space of our method is never larger than ten \( O(\lg^2 \ell) \) times the actual number of nodes on length \( \ell \) of the shortest paths.
Query Time Proportional to Path Length: Brief Sketch

increasing neighborhoods
find cycle sep. $C$

$|C| = O(tw)$
Outline

- Introduction
- Planar Separators
- Approximate Distance Oracles
- Exact Distance Oracles
- Conclusion
Contributions and Outlook

- Query Time (quasi-) proportional to Shortest-Path Length (almost matches experimental results)

- Application determines how much space $S \geq m$, our tradeoffs tell how to use it for fast query time $Q$ both for approximate and exact

- Main open question: optimal use?
  - Exact $S \cdot Q \geq n \sqrt{n}$ ?
  - $(1 + \epsilon)$–Approximate $S \cdot Q \geq n \lg n$ ?
  - $O(1)$–Approximate $S \cdot Q \geq n \lg \lg n$ ?

Planar spanners and approximate shortest path queries among obstacles in the plane.


Ittai Abraham and Cyril Gavoille.

Object location using path separators.

Details in LaBRI Research Report RR-1394-06.

Reinhard Bauer and Daniel Delling.

SHARC: Fast and robust unidirectional routing.


Reinhard Bauer, Daniel Delling, Peter Sanders, Dennis Schieferdecker, Dominik Schultes, and Dorothea Wagner.

Combining hierarchical and goal-directed speed-up techniques for Dijkstra’s algorithm.
Holger Bast, Stefan Funke, Peter Sanders, and Dominik Schultes.  
Fast routing in road networks with transit nodes.  

Sergio Cabello.  
Many distances in planar graphs.  
A preprint of the journal version is available in the University of Ljubljana preprint series, Vol. 47 (2009), 1089.

Danny Z. Chen and Jinhui Xu.  
Shortest path queries in planar graphs.  

Youssou Dieng and Cyril Gavoille.  
On the tree-width of planar graphs.  
Hristo Djidjev.
Efficient algorithms for shortest path problems on planar digraphs.

Jittat Fakcharoenphol and Satish Rao.
Planar graphs, negative weight edges, shortest paths, and near linear time.

Greg N. Frederickson.
Fast algorithms for shortest paths in planar graphs, with applications.

Andrew V. Goldberg and Chris Harrelson.
Computing the shortest path: A* search meets graph theory.


Peter Sanders and Dominik Schultes. Highway hierarchies hasten exact shortest path queries.
Robert Sedgewick and Jeffrey Scott Vitter.
Shortest paths in euclidean graphs.
Announced at FOCS 1984.

Mikkel Thorup.
Compact oracles for reachability and approximate distances in planar digraphs.