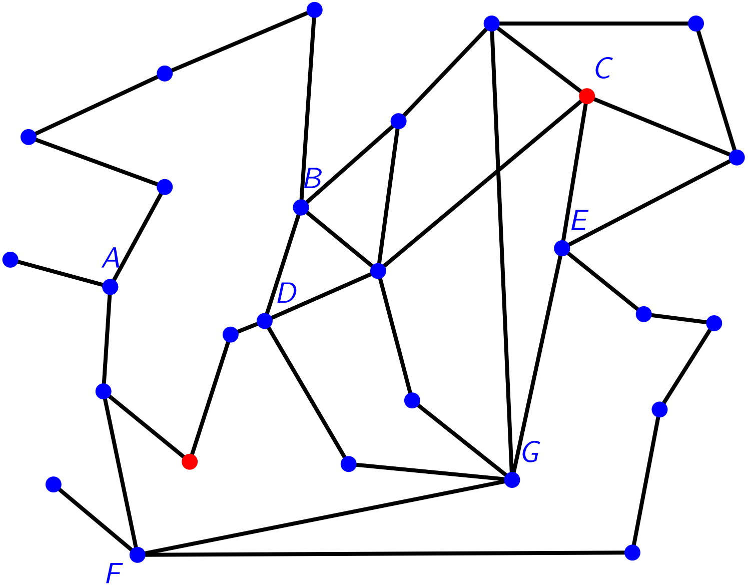
Linear-Space Approximate Distance Oracles for Planar, Bounded-Genus, and Minor-Free Graphs

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

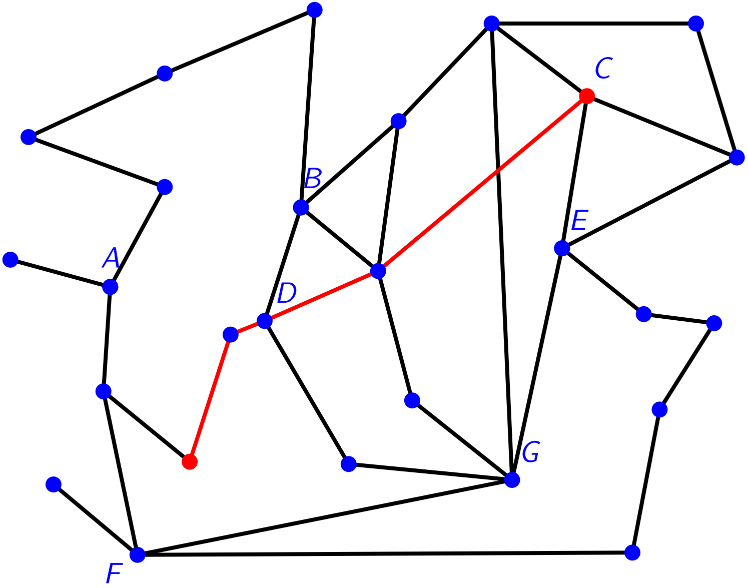
Joint work with Ken-ichi Kawarabayashi and Philip N. Klein

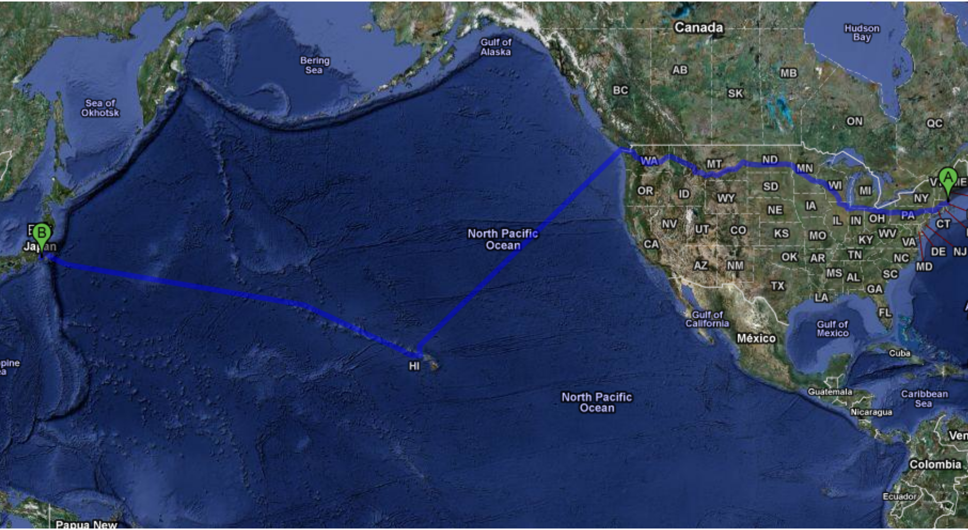
4 July 2011, ICALP

Motivation



Motivation




Canada
Hudson Bay
Gulf of Alaska
Bering Sea
Sea of Okhotsk
AB
MB
BC
SK
ON
QC
WA
MT
ND
MN
OR
ID
WY
SD
WI
MI
NY
A
B
NE
IA
NV
UT
CO
KS
MO
IL
IN
OH
PA
CT
CA
WV
KY
VA
DE
NJ
OK
AR
TN
NC
MD
AZ
NM
MS
AL
SC
TX
GA
LA
FL
North Pacific Ocean
Gulf of California
Gulf of Mexico
México
Cuba
HI
Guatemala
Caribbean Sea
Nicaragua
North Pacific Ocean
Colombia
Ecuador
Papua New

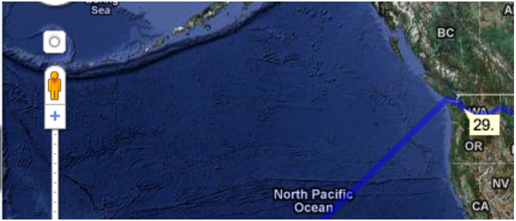
27. Turn left at **6th Ave NE** — 79 ft

28. Turn right at **NE Northlake Way** — 1.0 mi

29. Kayak across **the Pacific Ocean** — 2,756 mi

30. Continue straight — 0.1 mi

31. Turn left at **Kuilima Dr** — 0.5 mi



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.
  - $\widetilde{d}(u, v)$

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.
  - $\widetilde{d}(u, v)$
- Tradeoffs between Stretch, Space, and Query Time

Approximate Distance Oracles — Stretch

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

$$d_G(u, v) \leqslant \widetilde{d}(u, v) \leqslant (1 + \epsilon) \cdot d_G(u, v).$$

- Multiplicative Stretch $1 + \epsilon$

Related Work

Practical

- Focus on Transportation Networks

Theoretical

- General, undirected graphs
- Restricted classes (planar, bounded tree-width, bounded genus, 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]
- Methods that work very well for road networks (separators)
- $\rightsquigarrow$ see also Session A9, Wednesday @12:00

Related Work

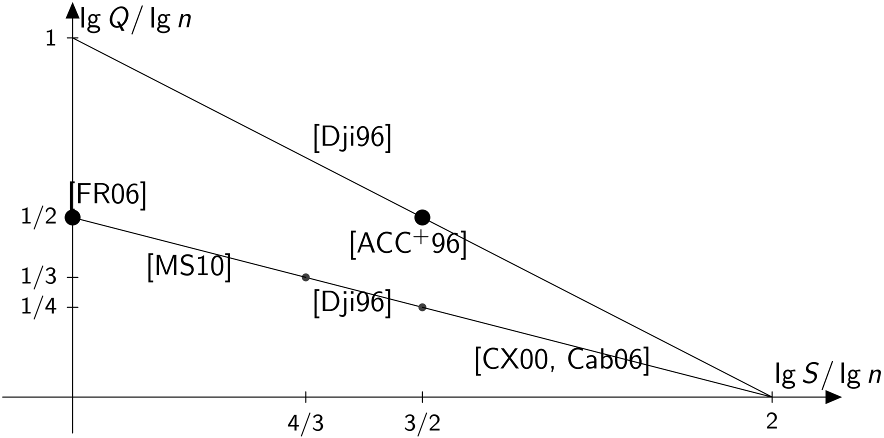
Practical

- Focus on Transportation Networks

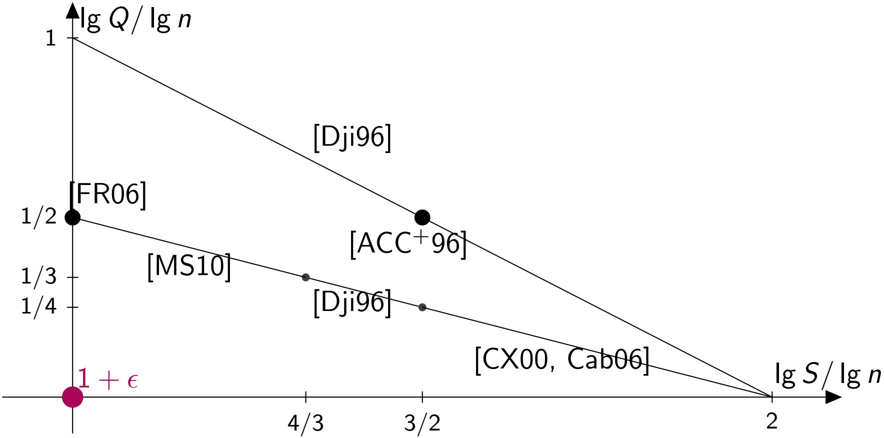
Theoretical

- General, undirected graphs
  large stretch or large space, or long query time
- Restricted graph classes (planar, small tree-width, bounded genus, minor-closed,...)
- $\rightsquigarrow$ see also Session A6, Tuesday @11:00

Space vs. Query Time for Exact Shortest Paths



Space vs. Query Time for Exact Shortest Paths



State of the Art

Approximate Distance Oracles for
Planar/Bounded-Genus/Minor-free Graphs

Efficient preprocessing	$O(n\epsilon^{-2}\log^3 n)$	$O(poly(n, \epsilon^{-1}))$
Quasilinear space	$O(n\epsilon^{-1}\log n)$	$O(n\epsilon^{-1}\log n)$
Fast query time	$O(\epsilon^{-1})$	$O(\epsilon^{-1}\log n)$
	Planar [Tho04]	Minor-free [AG06]

State of the Art and **Results**

Approximate Distance Oracles for
Planar/Bounded-Genus/Minor-free Graphs

Prepro	$O(n\epsilon^{-2}\log^3 n)$	$O(n\epsilon^{-2}\log^3 n)$	$O(poly(n,\epsilon^{-1}))$
Space	$O(n\epsilon^{-1}\log n)$	$O(n\epsilon^{-1}(g+\log n))$	$O(n\epsilon^{-1}\log n)$
Query	$O(\epsilon^{-1})$	$O(g\epsilon^{-1})$	$O(\epsilon^{-1}\log n)$
	Planar [Tho04]	Genus g	Minor-free [AG06]

Outline

$(1+\epsilon)$–Approximate Shortest-Path Queries; Planar G

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

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

Planar Separators, Graph $G = (V, E)$

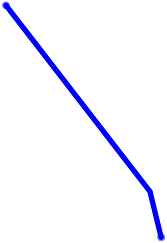
Partition V into V_1, V_2, S
such that $|V_1|, |V_2| \leqslant \frac{n}{2}$, no edge between V_1, V_2, and

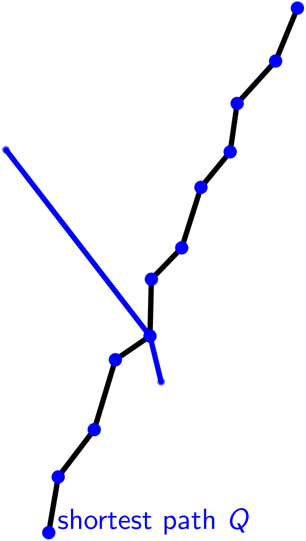
- Lipton and Tarjan [LT80], Miller [Mil86]
  - s.t. $|S| = \mathcal{O}(\sqrt{n})$
  - Shortest paths may cross S up to $\mathcal{O}(\sqrt{n})$ times
- Thorup [Tho04]
  - s.t. S consists of 3 shortest paths
  - can be extended to minor-closed families [AG06]
- Dieng and Gavoille [DG09]
  - s.t. S consists of $\mathcal{O}(1)$ shortest paths of length "tree-length"

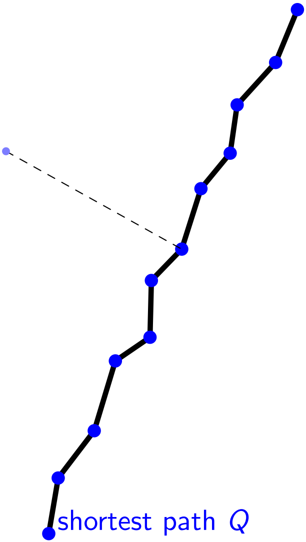
Main Techniques of Thorup's Distance Oracle

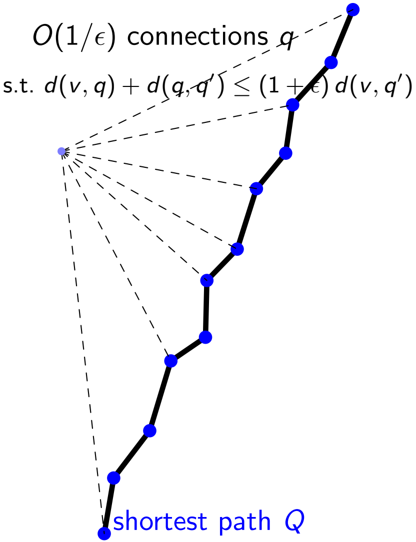
- Partition V into V_1, V_2, S such that $|V_1|, |V_2| \leqslant \frac{n}{2}$ and
  - s.t. S consists of 3 shortest paths Q
  - $\rightsquigarrow$ shortest paths cannot cross many times

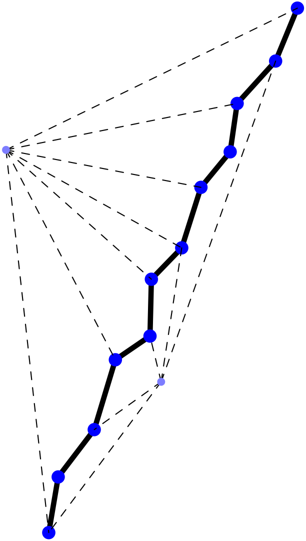
- Representation of paths that intersect Q




shortest path Q


shortest path Q


$O(1/\epsilon)$ connections q
s.t. $d(v,q) + d(q,q') \leq (1+\epsilon)\,d(v,q')$
shortest path Q



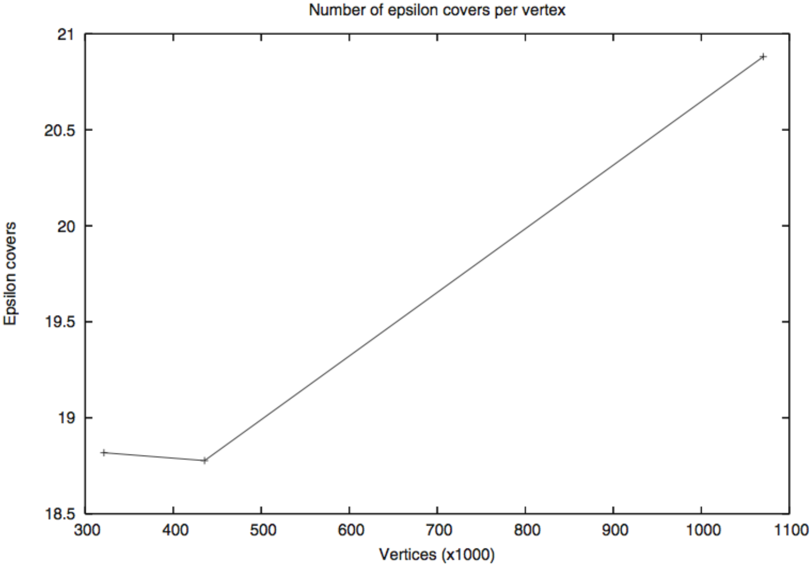
Space Consumption of Thorup's Distance Oracle

- 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

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
- "Reasonable" values:
  - $n \approx 10^7$, $\rightsquigarrow \log_2 n \approx 20$

Experimental Results [MZ07]



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
- "Reasonable" values:
  - $n \approx 10^7$, $\rightsquigarrow \log_2 n \approx 20$
  - $\epsilon = x\%$
  - $O(\cdot)$–constants quite small BUT 20 GBs for $\epsilon = 1\%$ [MZ07]

The total number of connections constructed during preprocessing gives an indication of the memory consumption of the oracle. In the current implementation, one connection consists of two floats, which is 8 bytes in total. The total number of connections for the *FLA* instance with $\epsilon = 0.01$ is around 250 million, which results in a memory consumption of about $2GB$. The number of connections is strongly affected by ϵ. When ϵ is increased to 0.10 for the same instance, the number of connections drops to just under 100 million.

Too much for mobile devices?


0.90
A1 Montrose Terrace

$(1+\epsilon)$–Approximate Shortest-Path Queries; Planar G

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

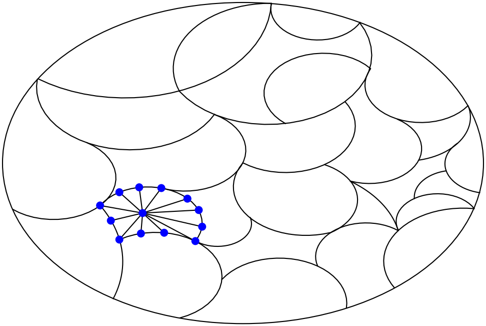
Assumption for this table: largest integer weight $N = O(poly(n))$
(complexity of oracles for planar digraphs depends on N)

Outline

Linear-Space Distance Oracle: Main Idea

- store connections for few nodes (landmarks)
  $\rightsquigarrow$ boundary of r–division

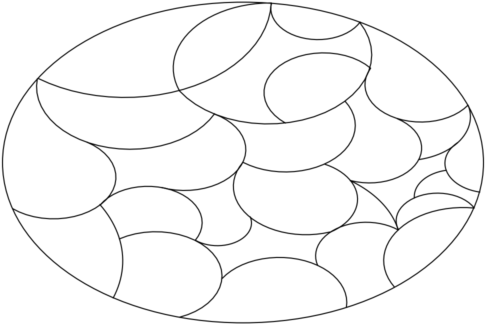
- at query time, search landmark then use [Tho04]

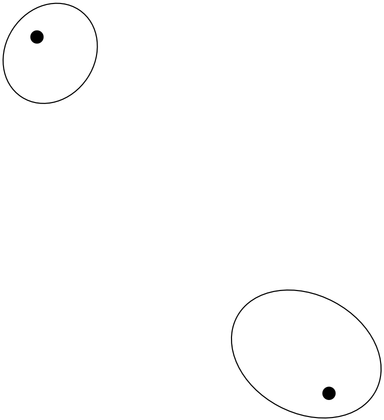


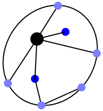
r–divisions [Fre87]

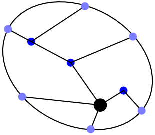
separate recursively (e.g. using [Mil86]) into

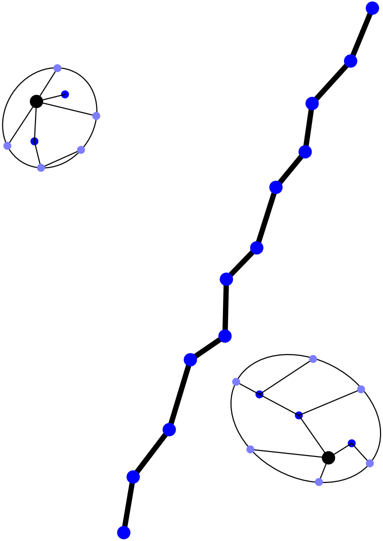
- $O(n/r)$ regions
- region size $O(r)$
- region boundary $O(\sqrt{r}) \rightsquigarrow$ total boundary $O(n/\sqrt{r})$

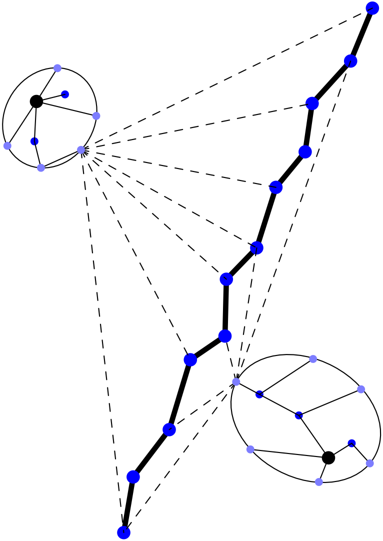


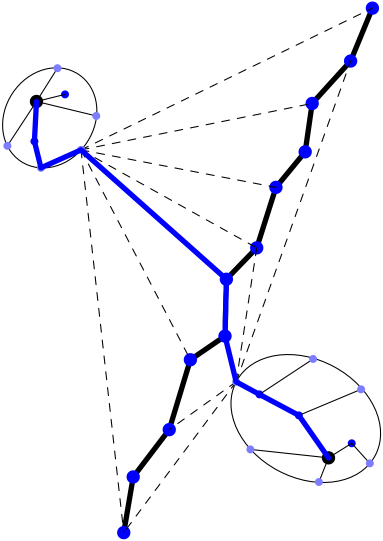


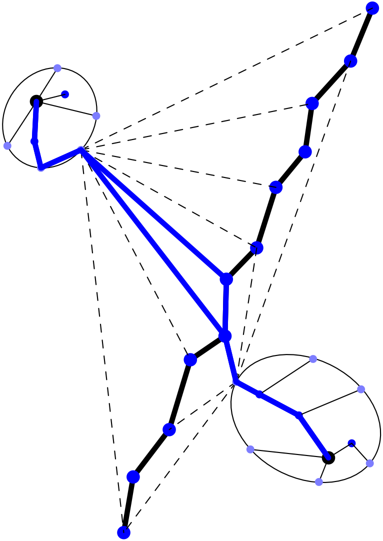












Distance Oracle: Space vs. Query Time

- Space: store connections for boundary of r–division
  $\rightsquigarrow O(\frac{n}{\sqrt{r}} \cdot \epsilon^{-1} \log n)$

  Linear Space: $\sqrt{r} = \epsilon^{-1} \log n$

- Query time:
  - search landmark $O(r)$ [HKRS97]
  - merge $O(\sqrt{r} \cdot \epsilon^{-1} \log n)$ connections [Tho04]

Distance Oracle: Space vs. Query Time

- Space: store connections for boundary of r–division
  $\rightsquigarrow O(\frac{n}{\sqrt{r}} \cdot \epsilon^{-1} \log n)$

  Linear Space: $\sqrt{r} = \epsilon^{-1} \log n$
  Sublinear Additional Space: $\sqrt{r} \gg \epsilon^{-1} \log n$

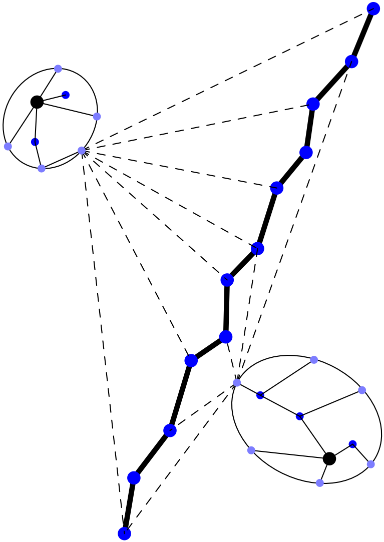
- Query time:
  - search landmark $O(r)$ [HKRS97]
    recursion for large r
  - merge $O(\sqrt{r} \cdot \epsilon^{-1} \log n)$ connections [Tho04]

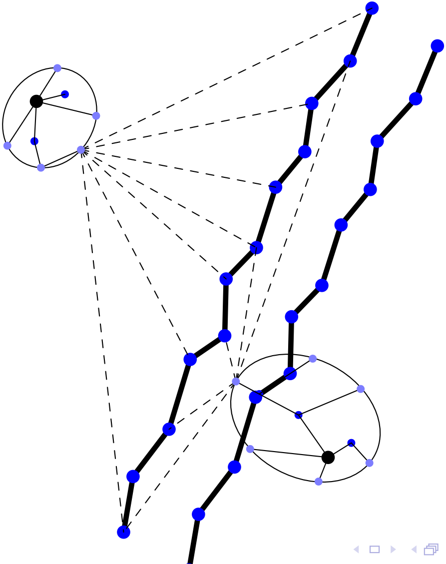
Distance Oracle: Space vs. Query Time

- Space: store connections for boundary of r–division
  $\rightsquigarrow O(\frac{n}{\sqrt{r}} \cdot \epsilon^{-1} \log n)$

  Linear Space: $\sqrt{r} = \epsilon^{-1} \log n$
  Sublinear Additional Space: $\sqrt{r} \gg \epsilon^{-1} \log n$

- Query time:
  - search landmark $O(r)$ [HKRS97]
    recursion for large r
  - merge $O(\sqrt{r} \cdot \epsilon^{-1} \log n)$ connections [Tho04]
    clever merge?





$(1+\epsilon)$–Approximate Shortest-Path Queries; Planar G

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

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

State of the Art and **Results**

Approximate Distance Oracles for
Planar/Bounded-Genus/Minor-free Graphs

Prepro	$O(n\epsilon^{-2}\log^3 n)$	$O(n\epsilon^{-2}\log^3 n)$	$O(poly(n,\epsilon^{-1}))$
Space	$O(n\epsilon^{-1}\log n)$	$O(n\epsilon^{-1}(g+\log n))$	$O(n\epsilon^{-1}\log n)$
Query	$O(\epsilon^{-1})$	$O(g\epsilon^{-1})$	$O(\epsilon^{-1}\log n)$
	Planar [Tho04]	Genus g	Minor-free [AG06]

State of the Art and **Results**

Linear-Space Approximate Distance Oracles for Planar/Bounded-Genus/Minor-free Graphs

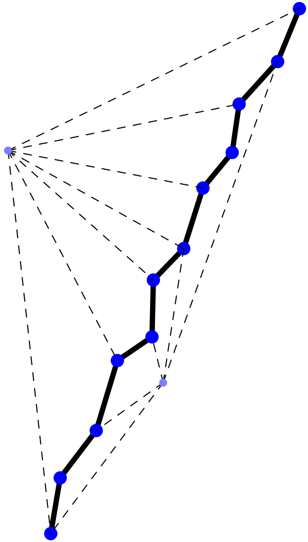
Prepro	$O(n \log^2 n)$	$O(n(g^3 + \log n) \log n)$	$O(poly(n, \epsilon^{-1}))$
Space	$O(n)$	$O(n)$	$O(n)$
Query	$O(\epsilon^{-2} \log^2 n)$	$O(\epsilon^{-2}(g + \log n)^2)$	$O(\epsilon^{-2} \log^2 n)$
	Planar	Genus g	Minor-free

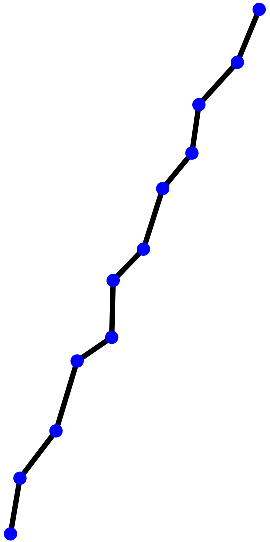
Outline

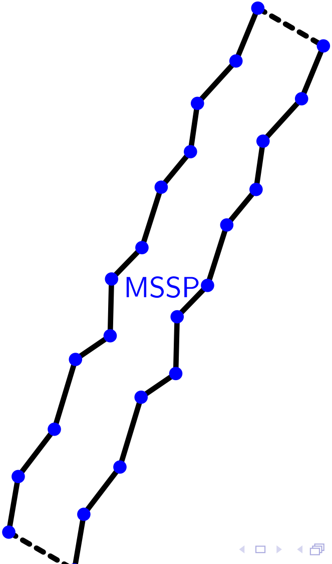
Preprocessing Improvements

[Tho04] $n\epsilon^{-2}\log^4 n \rightsquigarrow$ NEW $n\log^2 n$

- Use preprocessing for slower query time implicit in [Tho04], $n\epsilon^{-1}\log^3 n$
- Main idea: compute connections only for boundary nodes
  Numbers:
  - amortized $\log^2 n$ per connection?
  - only n connections $\rightsquigarrow n\log^2 n$ total time?
- Proof: Can be done efficiently using dynamic trees as in [Kle05]

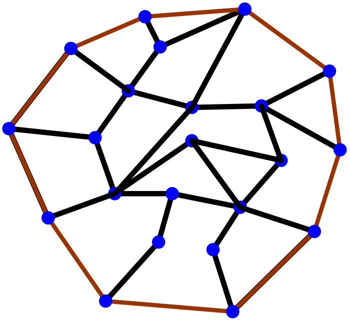




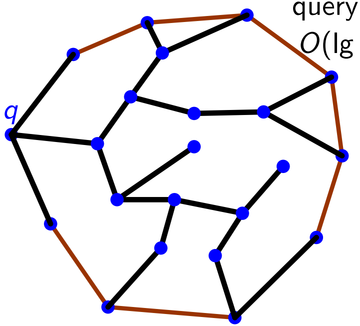

MSSP

Klein's MSSP data structure [Kle05]

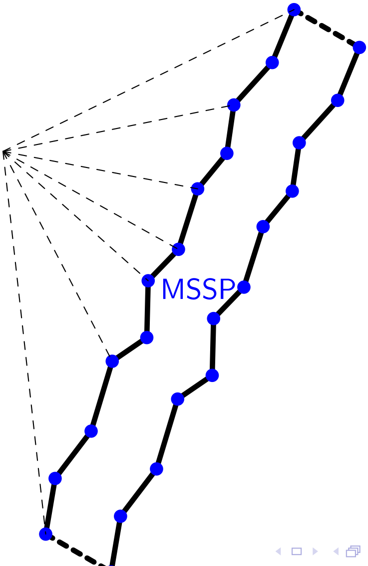
preprocess G, f
$O(n \lg n)$ time & space

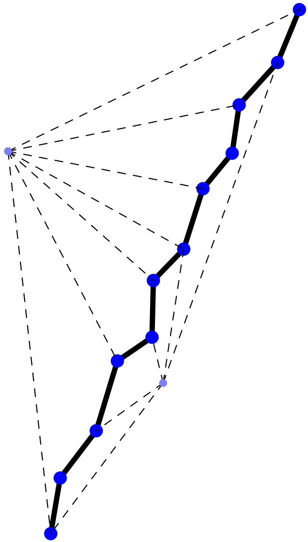


Klein's MSSP data structure [Kle05]



query $d(q, v)$
$O(\lg n)$ time


MSSP



Outline

Contributions and Outlook

- (Sub-)Linear-Space Approximate Distance Oracle
  Fast Preprocessing

- Application determines how much space $S \geq m$,
  our tradeoffs tell how to use it for fast query time Q
  (similar result for exact [MS10])

- Main open theory question: optimal use?

Exact	$S \cdot Q \geq n\sqrt{n}$	?
$(1+\epsilon)$–Approximate	$S \cdot Q \geq n \lg n$	?
$(1+\epsilon)$ & Unweighted	$S \cdot Q \leq n \lg \lg n \lg \lg \lg n$	!

Srinivasa Rao Arikati, Danny Z. Chen, L. Paul Chew, Gautam Das, Michiel H. M. Smid, and Christos D. Zaroliagis.
Planar spanners and approximate shortest path queries among obstacles in the plane.
In Algorithms - ESA '96, Fourth Annual European Symposium, Barcelona, Spain, September 25-27, 1996, Proceedings, pages 514–528, 1996.

Ittai Abraham and Cyril Gavoille.
Object location using path separators.
In Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006, pages 188–197, 2006.
Details in LaBRI Research Report RR-1394-06.

Reinhard Bauer and Daniel Delling.
SHARC: Fast and robust unidirectional routing.
In Proceedings of the 10th Workshop on Algorithm Engineering and Experiments (ALENEX'08), pages 13–26, 2008.

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.

In Experimental Algorithms, 7th International Workshop (WEA'08), Provincetown, MA, USA, May 30-June 1, 2008, Proceedings, pages 303–318, 2008.

Holger Bast, Stefan Funke, Peter Sanders, and Dominik Schultes.
Fast routing in road networks with transit nodes.
Science, 316(5824):566, 2007.

Sergio Cabello.
Many distances in planar graphs.
In Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006, pages 1213–1220, 2006.
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.
In Proceedings of the ACM Symposium on Theory of Computing (STOC), pages 469–478, 2000.

Youssou Dieng and Cyril Gavoille.
On the tree-width of planar graphs.
Electronic Notes in Discrete Mathematics, 34:593–596, 2009.

European Conference on Combinatorics, Graph Theory and Applications (EuroComb 2009).

Hristo Djidjev.
Efficient algorithms for shortest path problems on planar digraphs.
In Graph-Theoretic Concepts in Computer Science, 22nd International Workshop, WG '96, Cadenabbia (Como), Italy, June 12-14, 1996, Proceedings, pages 151–165, 1996.

Jittat Fakcharoenphol and Satish Rao.
Planar graphs, negative weight edges, shortest paths, and near linear time.
Journal of Computer and System Sciences, 72(5):868–889, 2006.
Announced at FOCS 2001.

Greg N. Frederickson.
Fast algorithms for shortest paths in planar graphs, with applications.
SIAM Journal on Computing, 16(6):1004–1022, 1987.

Andrew V. Goldberg and Chris Harrelson.
Computing the shortest path: A* search meets graph theory.
In Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'05), Vancouver, British Columbia, Canada, January 23-25, 2005, pages 156–165, 2005.

Monika Rauch Henzinger, Philip Nathan Klein, Satish Rao, and Sairam Subramanian.
Faster shortest-path algorithms for planar graphs.
Journal of Computer and System Sciences, 55(1):3–23, 1997.
Announced at STOC 1994.

Philip Nathan Klein.
Multiple-source shortest paths in planar graphs.
In Symposium on Discrete Algorithms (SODA), pages 146–155, 2005.

Richard J. Lipton and Robert Endre Tarjan.
Applications of a planar separator theorem.
SIAM Journal on Computing, 9(3):615–627, 1980.
Announced at FOCS 1977.

Gary L. Miller.
Finding small simple cycle separators for 2-connected planar graphs.
Journal of Computer and System Sciences, 32(3):265–279, 1986.
Announced at STOC 1984.

Shay Mozes and Christian Sommer.
Exact distance oracles for planar graphs.
CoRR, abs/1011.5549, 2010.

Laurent Flindt Muller and Martin Zachariasen.
Fast and compact oracles for approximate distances in planar graphs.
In Algorithms - ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings, pages 657–668, 2007.

Peter Sanders and Dominik Schultes.
Highway hierarchies hasten exact shortest path queries.
In Algorithms - ESA 2005, 13th Annual European Symposium, Palma de Mallorca, Spain, October 3-6, 2005, Proceedings, pages 568–579, 2005.

Robert Sedgewick and Jeffrey Scott Vitter.
Shortest paths in Euclidean graphs.
Algorithmica, 1(1):31–48, 1986.
Announced at FOCS 1984.

Mikkel Thorup.
Compact oracles for reachability and approximate distances in planar digraphs.
Journal of the ACM, 51(6):993–1024, 2004.
Announced at FOCS 2001.