Query Processing Using Distance Oracles for Spatial Networks

Xiaofei Zhao and Saladi Rahul

September 19, 2011
1 Introduction
Location based services are popular these days!

Shortest-path and nearest neighbour queries are very crucial.

Need for real-time execution of these spatial queries.
Shortest path query on a spatial network

- Preprocess a large spatial network so that given a source and a destination node \( s \) and \( d \), respectively, we need to find the shortest path along the network from \( s \) to \( d \).

- **Question** Given \( n \) vertices or nodes on the network, how can we find out the shortest distance between any two vertices ??
Previous solutions

- Precompute and store the shortest paths between all possible vertices!
  - Space can be reduced from $O(n^3)$ to $O(n^2)$.
- By exploiting spatial coherence of the destination vertices, Sankaranarayan et al., reduce the space to $O(n^{1.5})$. 
Objective of the work

To build an oracle of linear size for finding shortest distance. This is done by willing to expend:

- A bit more time to answer the query.
- Accept an error $\epsilon$ in the accuracy of the distance provided.
Major contributions

▶ Building a linear size oracle for finding shortest distance. Specifically of size $O\left(\frac{n}{\epsilon d}\right)$.

▶ Formulation of simple SQL constructs for answering spatial queries.
  ▶ Seamless integration into a relational database.

▶ Adaption of well-separated pair decomposition technique to a spatial network.

▶ *Bounded distortion* is independent of $n$. It lies between $(1 - \epsilon)$ and $(1 + \epsilon)$. 

Xiaofei Zhao and Saladi Rahul

Query Processing Using Distance Oracles for Spatial Networks
Key Idea

- $A$ is a set of source vertices close to each other.
- $B$ is a set of destination vertices close to each other.
- $A$ and $B$ are sufficiently far away from each other.
- Then the shortest path between them may share common vertices.
  - Implies that the network distance between any source vertex and any destination vertex will more or less be the same.
- This idea is captured by adapting the idea of well-separated pair decomposition.
Query processing made simple!

- **Major Challenge** Spatial networks cannot be easily represented using the relational model.

- Since network distance is approximated by one value, spatial queries can be formulated using appropriate SQL constructs and a few built-in primitives.

- **$\epsilon$-Approximate network distance**
  ```sql
  SELECT O.d_\epsilon FROM O WHERE O.Z_{AB} = Z_{4}(p, q).
  ```

- **Region Search**
  ```sql
  SELECT R.pos, O.d_\epsilon FROM R, O WHERE O.Z_{AB} = Z_{4}(q, R.pos) and R.type = "Italian" and O.d_\epsilon \leq 10 \text{ miles}.
  ```
Validation Methodology

- Asymptotic analysis of the space occupied and the query time. A nice trade-off between the space and distortion factor was set up.

- Experiments were also run to show that the techniques work in practice as well.

- A small case-study was also done to show how seamlessly these techniques can be integrated in a relational database.
Major assumptions in the paper

- Source and the destination vertices are *sufficiently far* apart.

- They *do not report the “shortest path”*. Only the network distance is reported.

- Updates to the road network were not considered.
Introduction

Improvements

- A lot of shortest path queries would involve source and destination pair being close to each other!
- Updates to the road network or handling of unexpected events
Thank You