USC Viterbi
School of Engineering

Query Processing in Spatial Network Databases

Instructor: Cyrus Shahabi

Outline

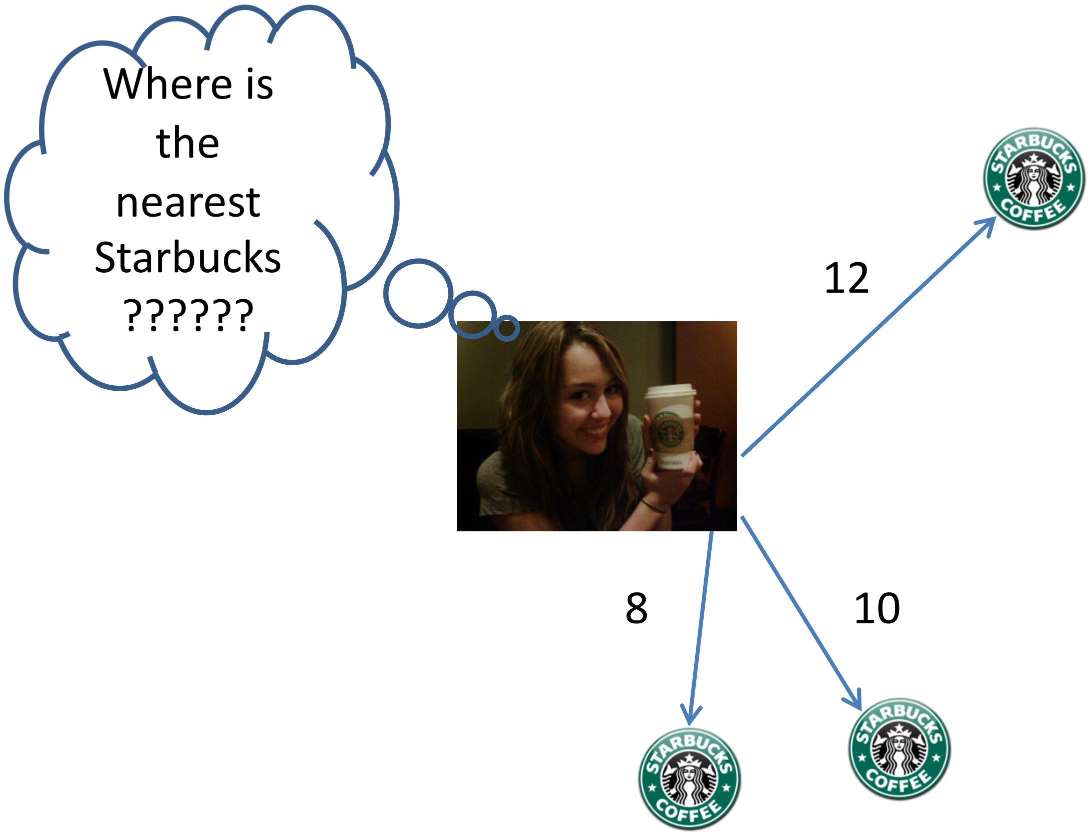
- Introduction
- Related work
  - Disk-based graph representations: 2DMatrix, CCAM structure
  - R-tree
- Spatial query in network databases
  - Architecture
  - Spatial queries:
    - Nearest neighbor query
      - Incremental Euclidean Restriction (IER) method
      - Incremental Network Expansion (INE) method
    - Range query
      - Range Euclidean Restriction (RER) method
      - Range Network Expansion (RNE) method
- Experiments
- Summary

Introduction

- Euclidean distance

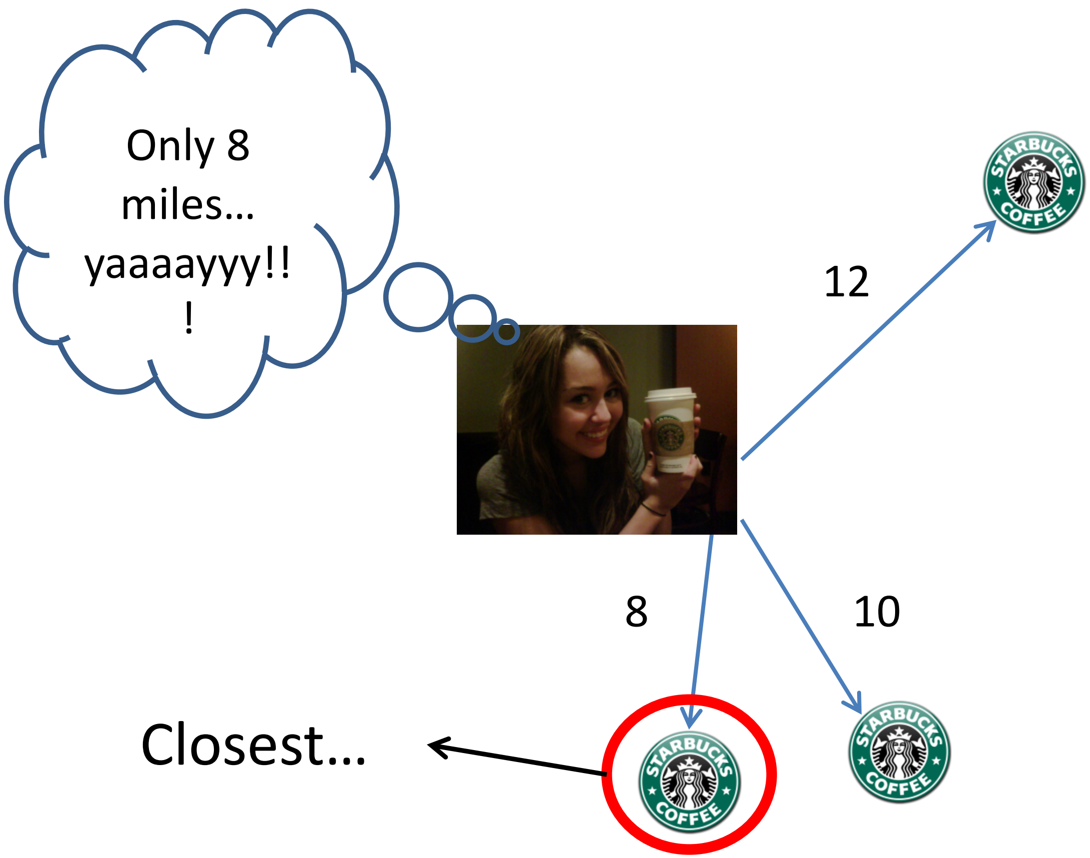
Introduction

- Euclidean distance



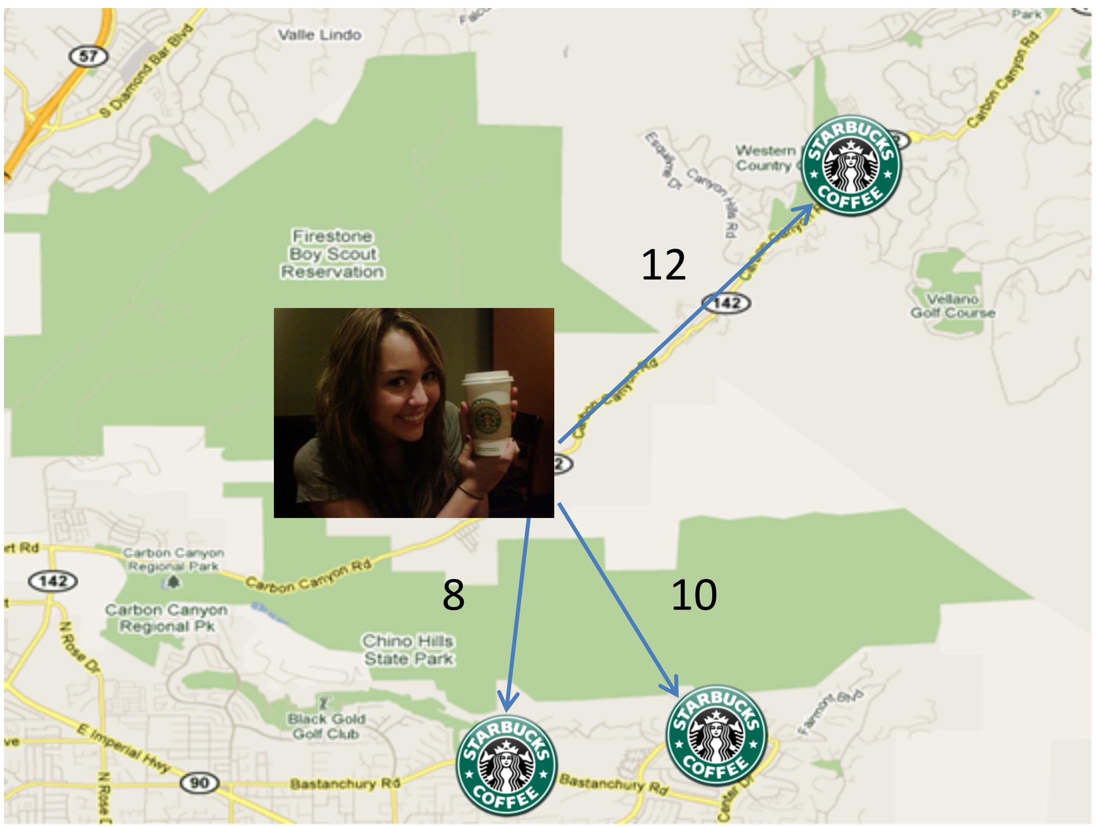
Introduction

- Euclidean distance



Introduction

- Network distance



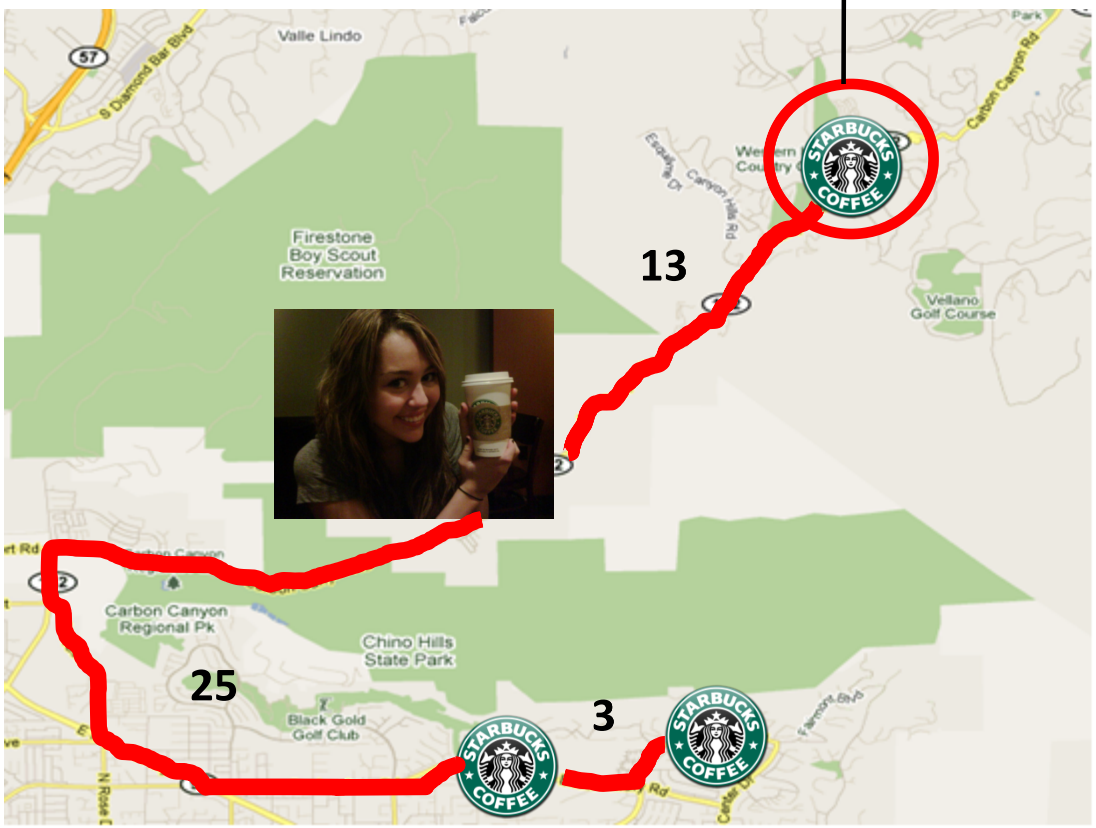
Considering the underlying road network, **is the nearest Starbucks still the same**

USC Viterbi
School of Engineering

Introduction

- Network distance

Closest…



NO !!!

Introduction

- Euclidean distance vs. Network distance

Any relationship?

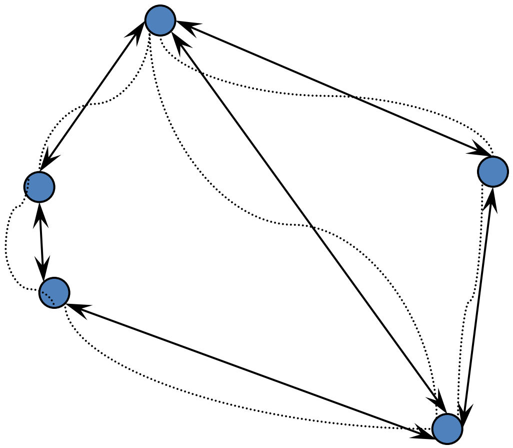
$$d_E \leq d_N$$

the Euclidean distance between two points is equal or smaller than their network distance.

USC Viterbi
School of Engineering

Introduction

- How can we represent a road network?
- Graphs



USC Viterbi
School of Engineering

Outline

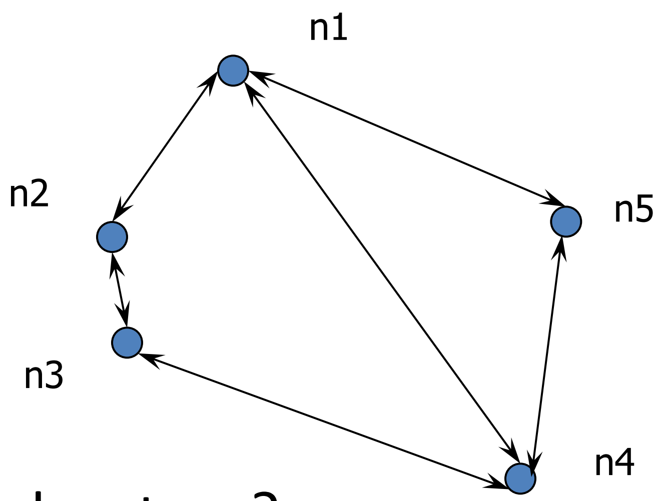
- Introduction
- Related work
  - Disk-based graph representations: 2DMatrix, CCAM structure
  - R-tree
- Spatial query in network databases
  - Architecture
  - Spatial queries:
    - Nearest neighbor query
      - Incremental Euclidean Restriction (IER) method
      - Incremental Network Expansion (INE) method
    - Range query
      - Range Euclidean Restriction (RER) method
      - Range Network Expansion (RNE) method
- Experiments
- Summary

Disk-based graph representations

- A graph can be represented as
  - Two-dimensional matrix
  - An adjacency list

Disk-based graph representations: 2D Matrix

– Two-dimensional matrix



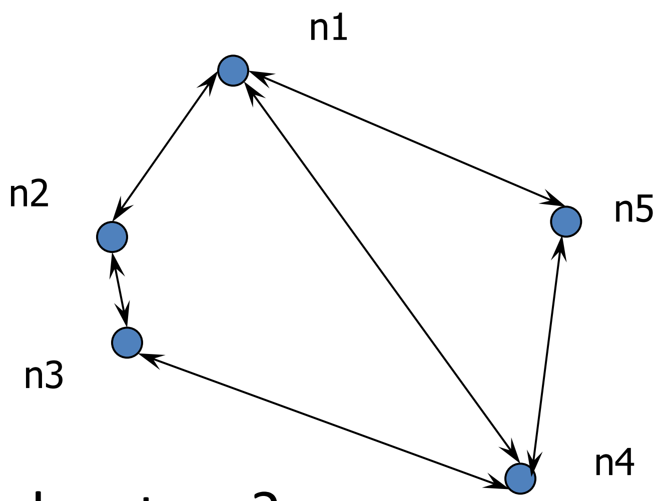
Adjacency matrix

	n1	n2	n3	n4	n5
n1	0	1	0	1	1
n2	1	0	1	0	0
n3	0	1	0	1	0
n4	1	0	1	0	1
n5	1	0	0	1	0

– Disadvantage?

Disk-based graph representations: 2D Matrix

– Two-dimensional matrix



Adjacency matrix

	n1	n2	n3	n4	n5
n1	0	1	0	1	1
n2	1	0	1	0	0
n3	0	1	0	1	0
n4	1	0	1	0	1
n5	1	0	0	1	0

– Disadvantage?

Sparse, More I/O

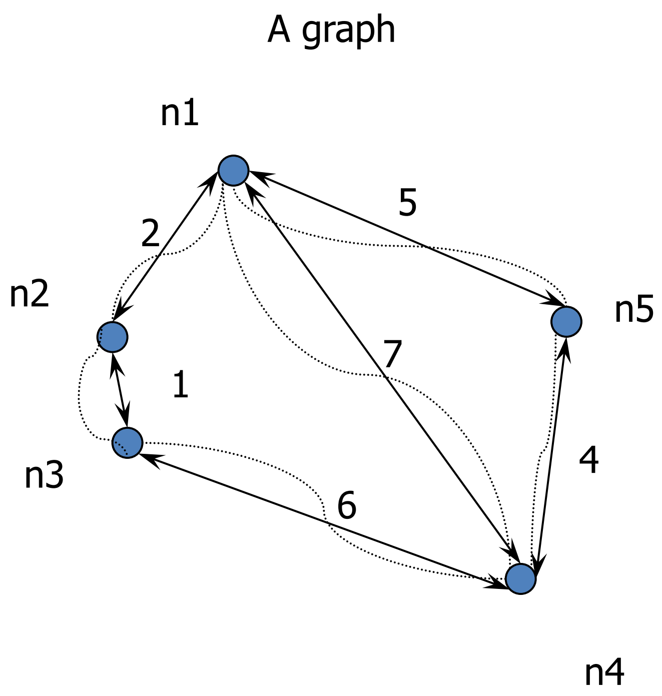
USC Viterbi
School of Engineering

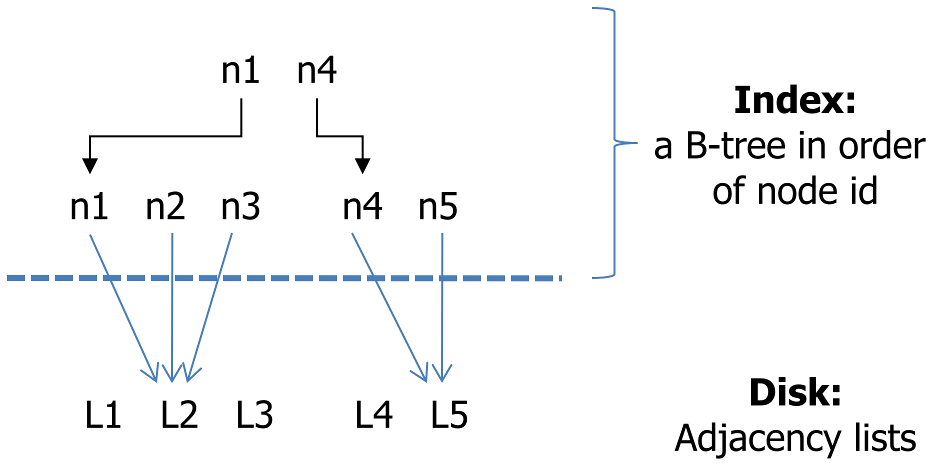
Disk-based graph representations: CCAM structure

- The Connectivity-clustered Access Method (CCAM) structure
  - Each node has a list that stores its neighbors
  - Stores the lists of neighbor nodes together

Disk-based graph representations: CCAM structure

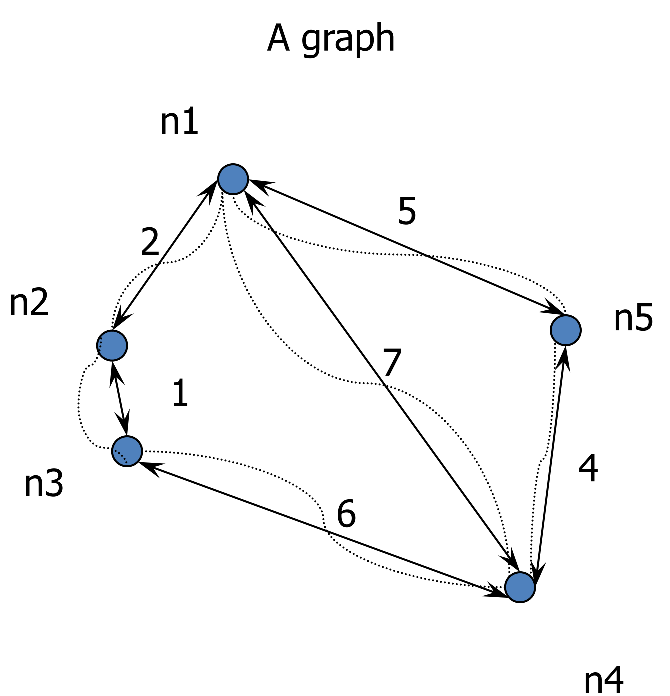
- An example

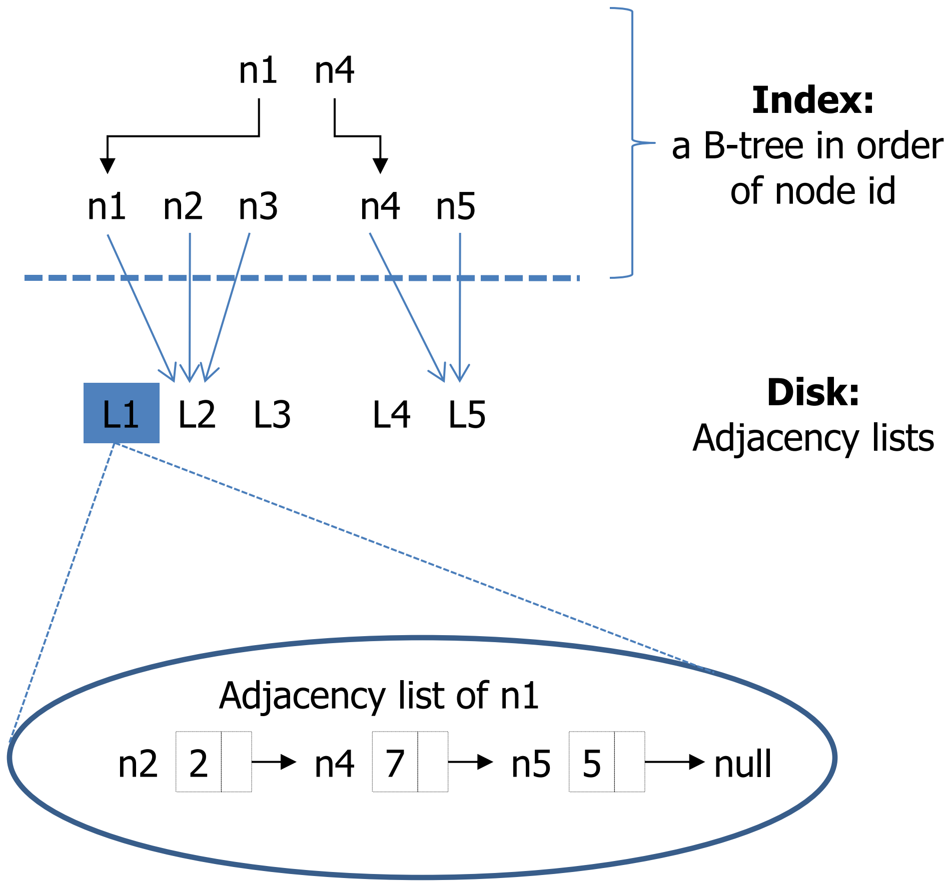




Disk-based graph representations: CCAM structure

- An example





USC Viterbi
School of Engineering

Disk-based graph representations- CCAM structure

- CCAM vs. 2D Matrix ?
- CCAM is preferable for applications, such as road networks, where the graphs are sparse.

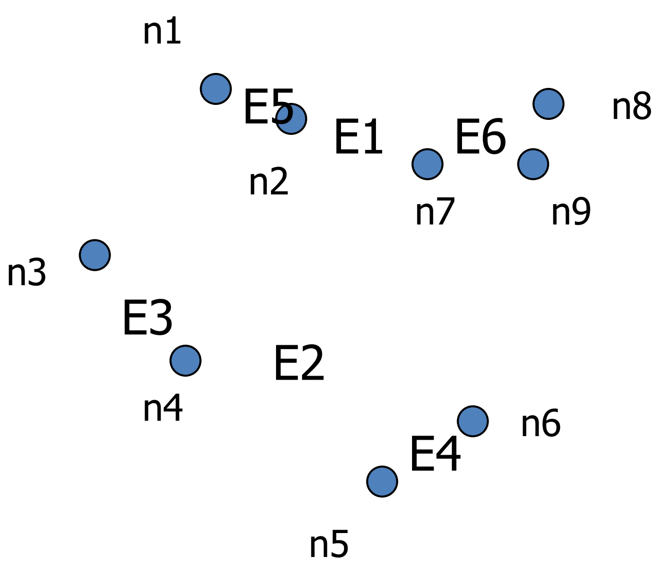
USC Viterbi School of Engineering

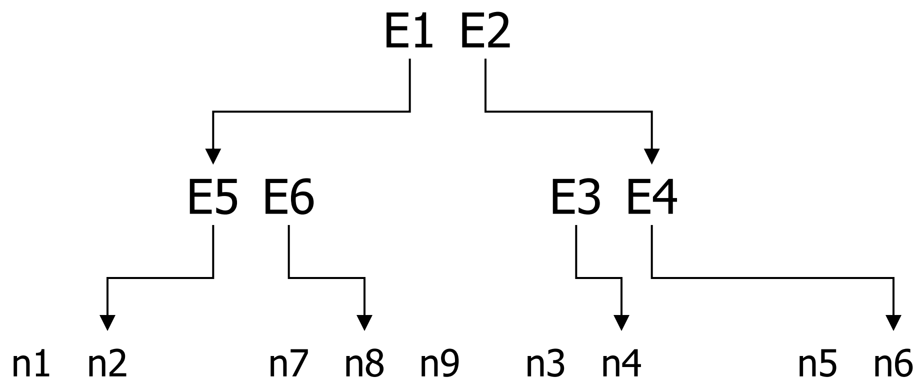
Outline

- Introduction
- Related work
  - Disk-based graph representations: 2DMatrix, CCAM structure
  - R-tree
- Spatial query in network databases
  - Architecture
  - Spatial queries:
    - Nearest neighbor query
      - Incremental Euclidean Restriction (IER) method
      - Incremental Network Expansion (INE) method
    - Range query
      - Range Euclidean Restriction (RER) method
      - Range Network Expansion (RNE) method
- Experiments
- Summary

Spatial query processing in Euclidean Space

- An R-tree index
  - Multidimentional extension of B-tree



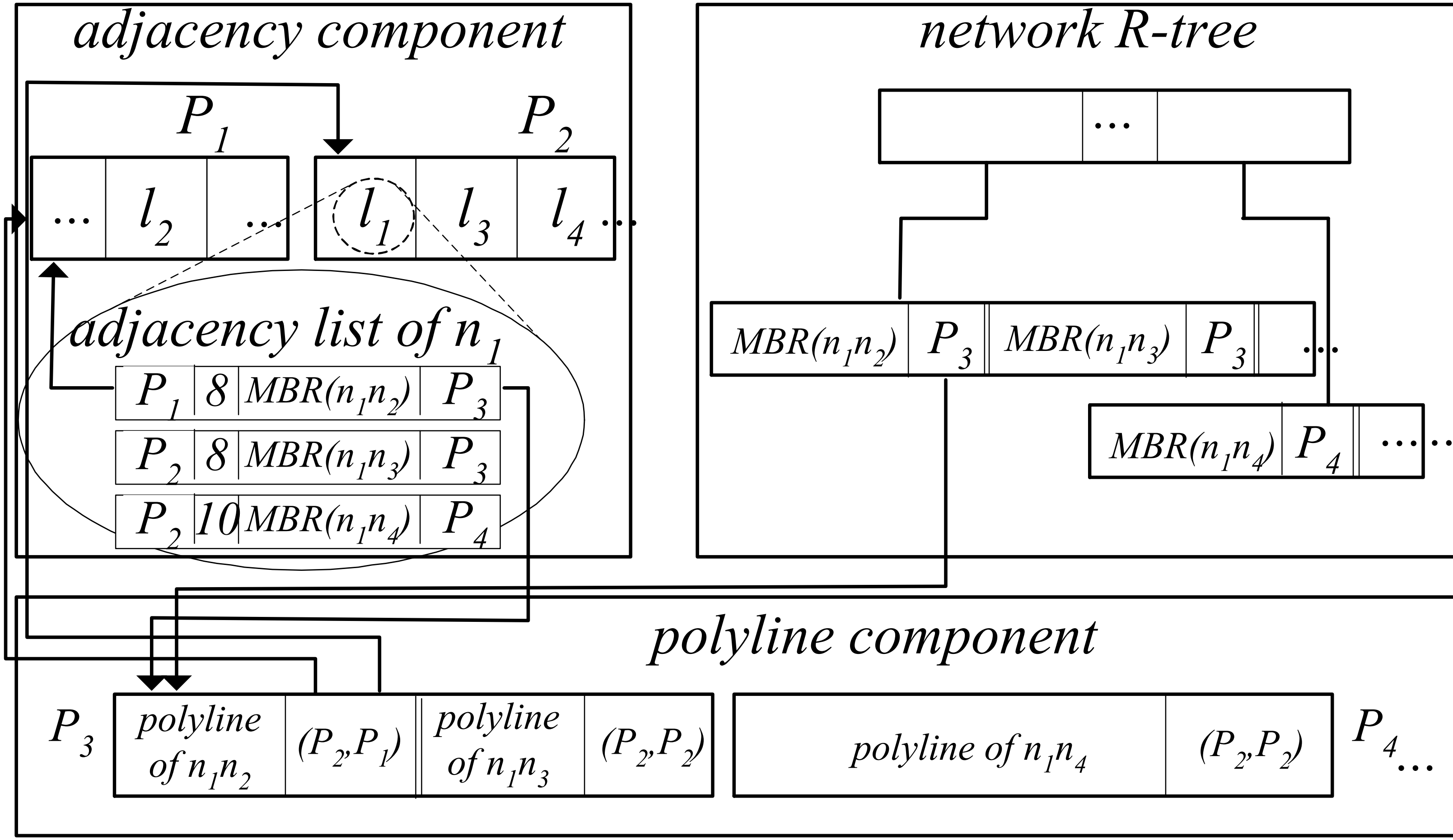


Outline

- Introduction
- Related work
  - Disk-based graph representations- 2DMatrix, CCAM structure
  - R-tree
- Spatial query in network databases
  - Architecture
  - Spatial queries:
    - Nearest neighbor query
      - Incremental Euclidean Restriction (IER) method
      - Incremental Network Expansion (INE) method
    - Range query
      - Range Euclidean Restriction (RER) method
      - Range Network Expansion (RNE) method
- Experiments
- Summary

Architecture

- Underlying network and spatial entities are separated
- Index the entity datasets (e.g., hotels) separately by R-trees (called Objects R-tree)
- For the network: preserve connectivity (adjacency comp) & location (network R-tree)



Functions

- *check_entity*(*seg*, *p*): is a Boolean function that returns true if point (entity) *p* lies on the network segment *seg* (i.e., seg covers p)
  - MBR of *seg* is used for filtering and its poly-line representation for refinement.

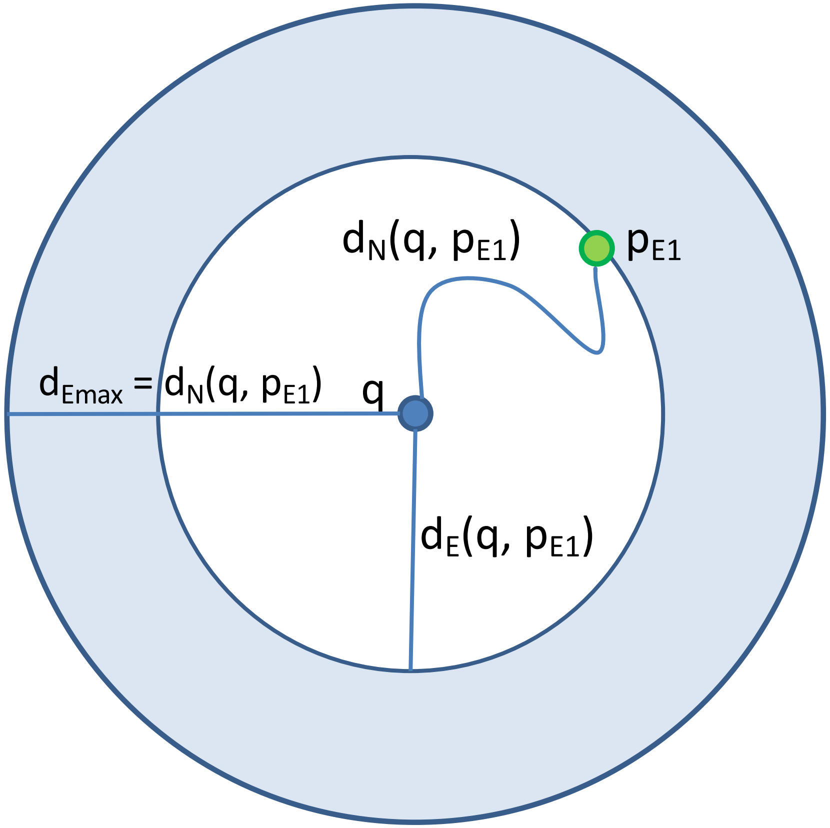
- *find_segment*(*p*): outputs the segment that covers point *p* by performing a point location query on the network R- tree.

- *find_entities*(*seg*): returns entities covered by segment *seg*.
  - first finds all the candidate entities that lie in the MBR of *seg*, and then eliminates the false hits using the poly-line of *seg*.

- *compute_ND*(*p1*,*p2*): returns the network distance of two arbitrary points *p1*, *p2* in the network,
  - by applying a (secondary-memory) Dijkstra's algorithm to compute the shortest path from *p1* to *p2*.

Nearest Neighbor - IER

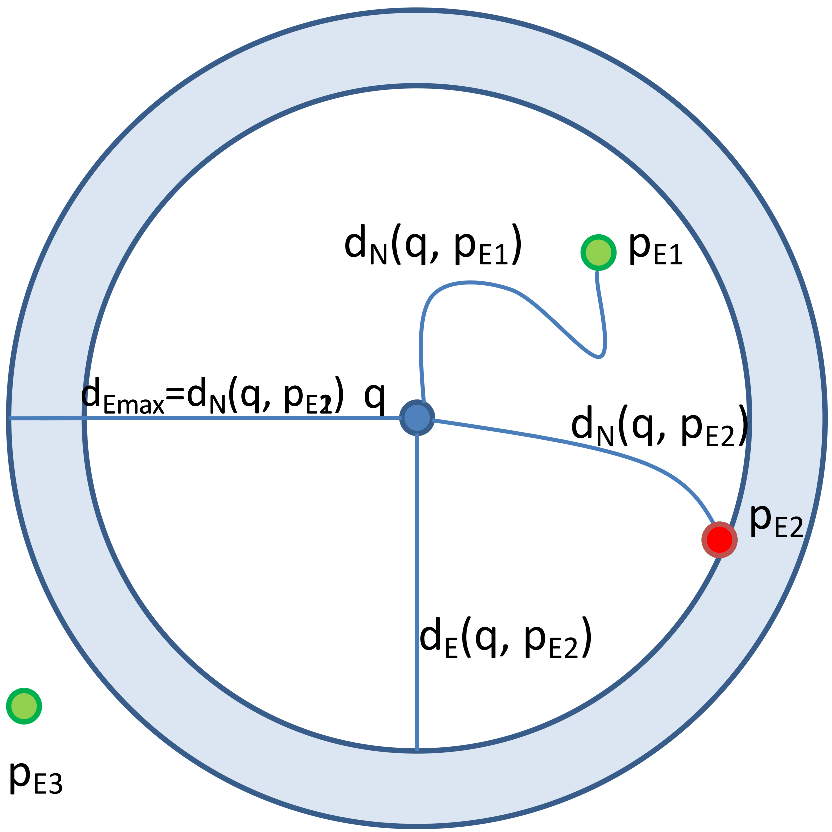
- Example: NN of q ?



- Step 1: Find Euclidean NN p_{E1} of q on the entity R-tree
- Step 2: Compute the network distance $d_N(q, p_{E1})$ of p_{E1}
- Step 3: **Euclidean lower-bound.** Objects closer to q than p_{E1} should be within Euclidean distance $d_{Emax} = d_N(q, p_{E1})$ from q. Only check SHADED AREA!

Nearest Neighbor - IER

- Example: NN of q ?

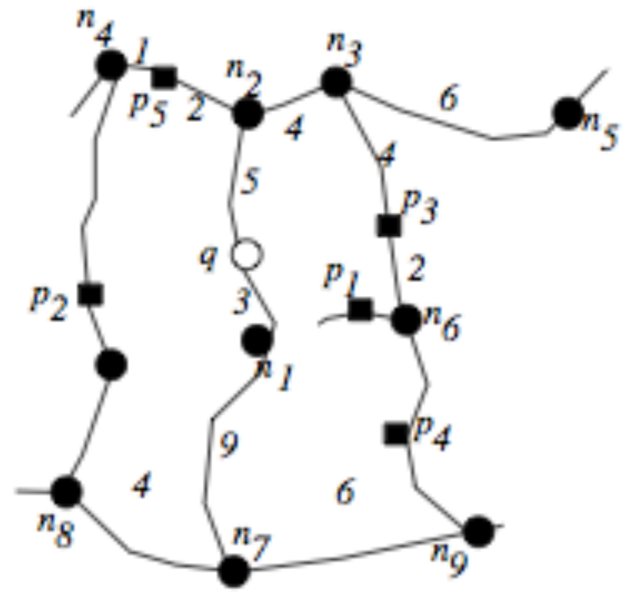


- Step 1: Find Euclidean NN p_{E1} of q on the entity R-tree
- Step 2: Compute the network distance $d_N(q, p_{E1})$ of p_{E1}
- Step 3: **Euclidean lower-bound.** Objects closer to q than p_{E1} should be within Euclidean distance $d_{Emax} = d_N(q, p_{E1})$ from q. Only interesting if in SHADED AREA!

- Step 4: Find 2[nd] Euclidean NN p_{E2} within range d_{Emax} .
- Step 5: $d_N(q, p_{E2}) < d_N(q, p_{E1})$. Current NN is p_{E2} .
- Step 6: Set $d_{Emax} = d_N(q, p_{E2})$. Range is now smaller.
- Step 7: Next NN p_{E3} is out of d_{Emax} range. Terminate!

IER: Shortcoming

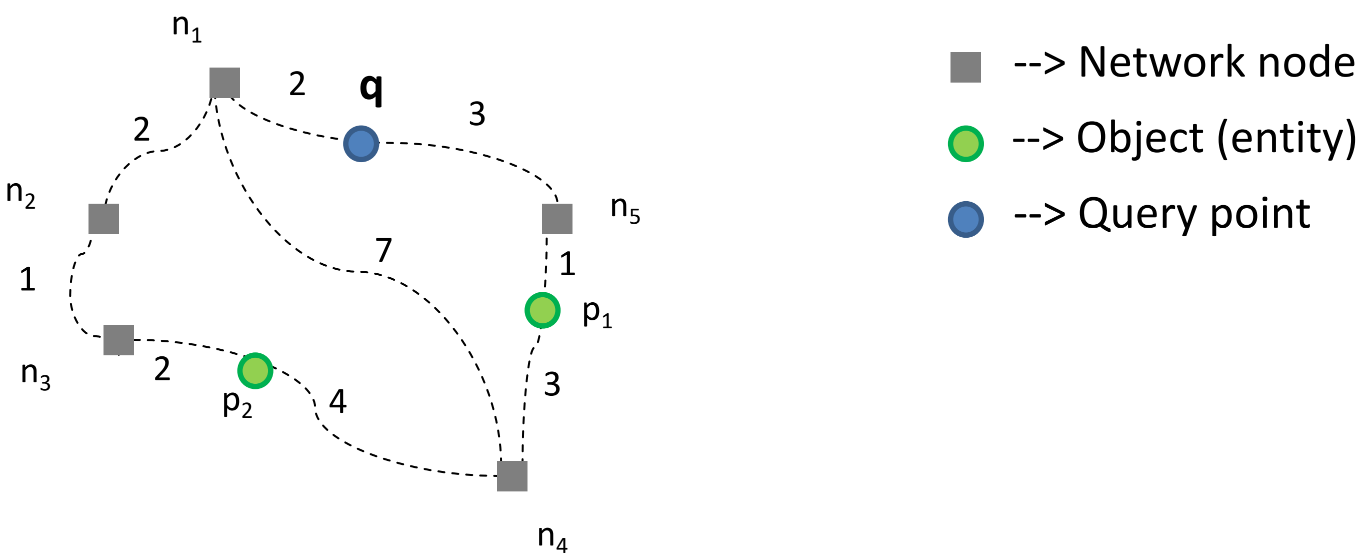
- IER performs well if the ranking of the data points by their Euclidean distance is similar to that with respect to the network distance. Otherwise, a large number of Euclidean NNs may be inspected before the network NN is found
- E.g., P5 is the closest in network distance but farther than p1 to p4 in Euclidean distance



Since *p5* has the largest Euclidean distance, it will be examined after all other entities, i.e., *p1* to *p4* correspond to *false hits*, for which the network distance computations are redundant.

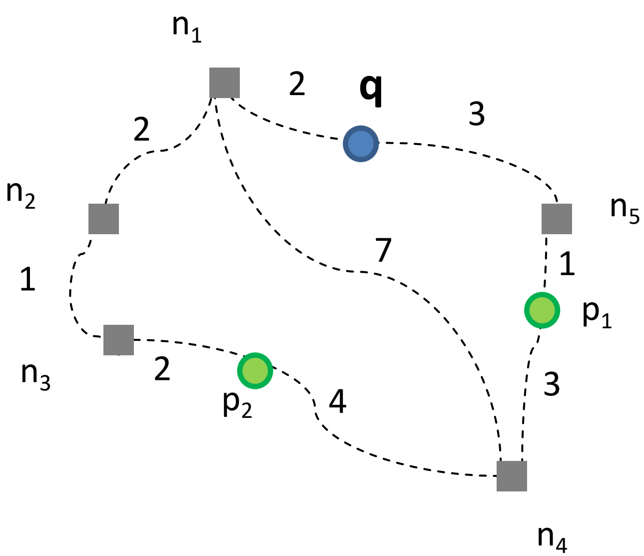
Nearest Neighbor – INE

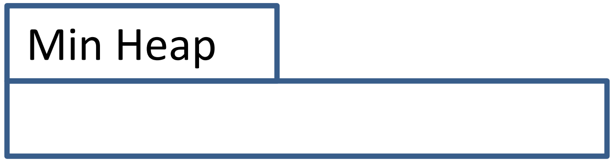
- Incremental Network Expansion (INE) performs network expansion (starting from *q*), and examines entities in the order they are encountered.



Nearest Neighbor – INE

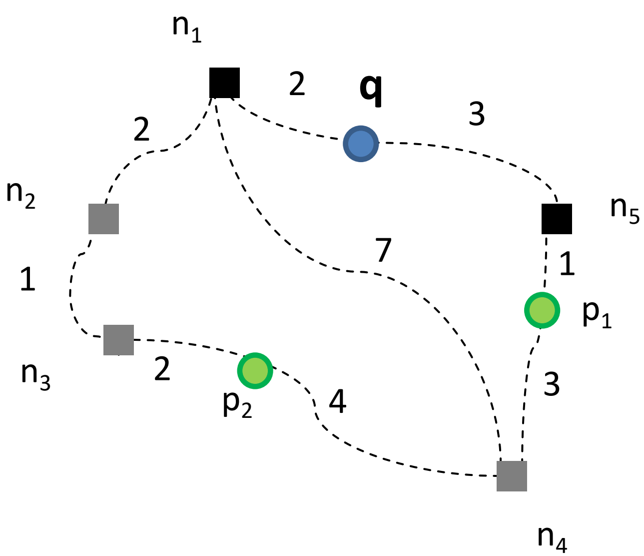
Find NN for a given query point q.





Nearest Neighbor – INE

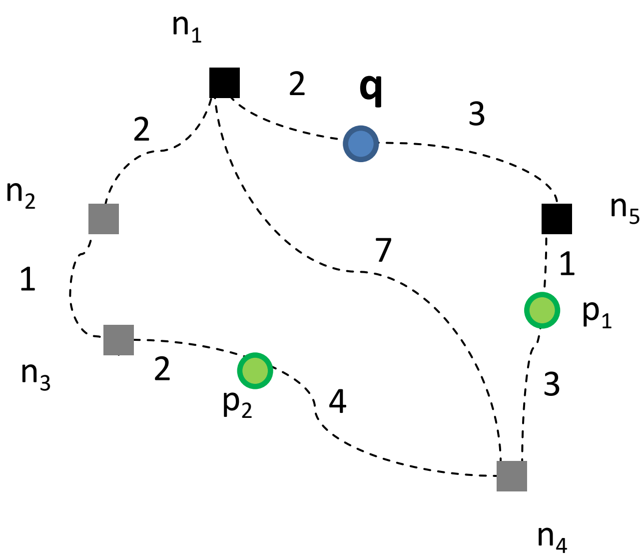
Find the edge that q is on using Network R-tree, initiate heap



Min Heap
$(n_1, 2), (n_5, 3)$

Nearest Neighbor – INE

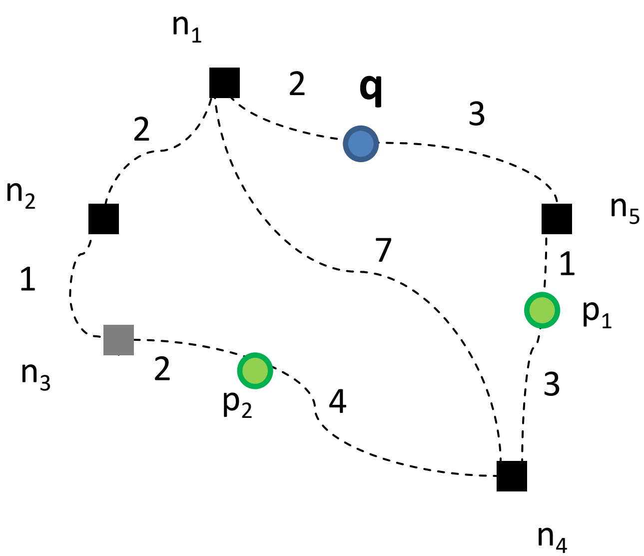
For each edge *e* in heap, find entities that are on *e* and add *e*'s neighboring edges into heap.



Min Heap
(n_1, 2), (n_5, 3)

Nearest Neighbor – INE

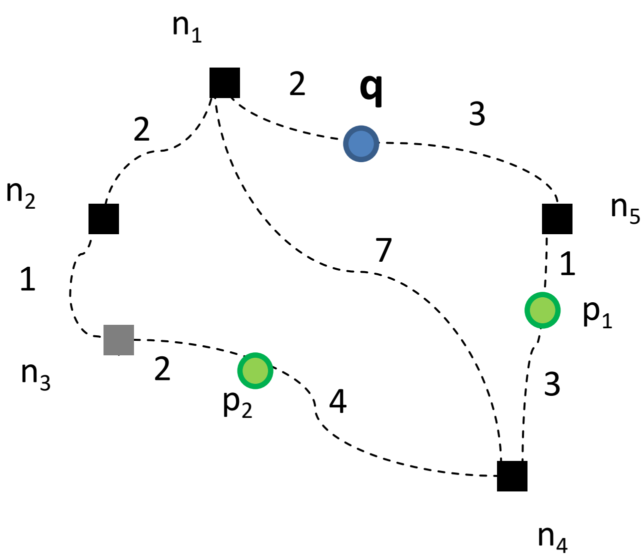
Add n_2 and n_4 into heap



Min Heap
$(n_5, 3), (n_2, 4), (n_4, 9)$

Nearest Neighbor – INE

Add n_4 into heap



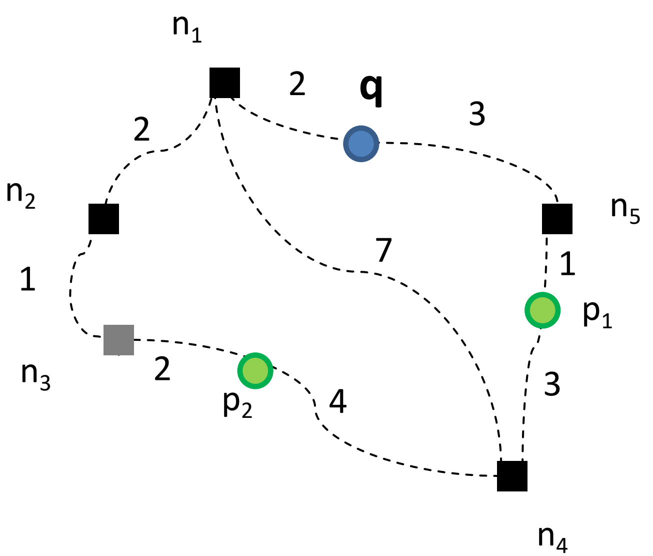
Min Heap

$(n_2, 4), (n_4, 7), \cancel{(n_4, 9)}$

USC Viterbi School of Engineering

Nearest Neighbor – INE

p_1 is found on the edge (n_5, n_4)



- $d_N(q, p_1) = d_{Nmax} = 4$
- The next entry in the heap n_2 doesn't have a smaller distance than d_{Nmax}. Thus, the algorithm **terminates**.

Min Heap
$(n_2, 4), (n_4, 7)$

USC Viterbi
School of Engineering

Outline

- Introduction
- Related work
  - Disk-based graph representations: 2DMatrix, CCAM structure
  - R-tree
- Spatial query in network databases
  - Architecture
  - Spatial queries:
    - Nearest neighbor query
      - Incremental Euclidean Restriction (IER) method
      - Incremental Network Expansion (INE) method
    - Range query
      - Range Euclidean Restriction (RER) method
      - Range Network Expansion (RNE) method
- Experiments
- Summary

USC Viterbi
School of Engineering

Range Queries – ER

- Range query: given a query point *q*, a range *e* and a spatial dataset *S*, find all objects that are within network distance *e from q*.
- <u>Range Euclidean Restriction (RER) method:</u>
  – Perform a range query at the entity dataset and find the set of objects *S'* within (Euclidean) distance *e* from *q*.
    - *S'* is guaranteed to avoid **false misses** using lower-bound property ($d_N(q,p) \leq e \Rightarrow d_E(q,p) \leq e$).
    - But it may contain a large number of ***false hits***.
  – RER examines all segments within network distance *e* from *q*. Points of *S'* that fall on some segment, are removed from *S'* and returned to the user as result.
  – The process terminates when all the segments in the range are exhausted, or when *S'* becomes empty.

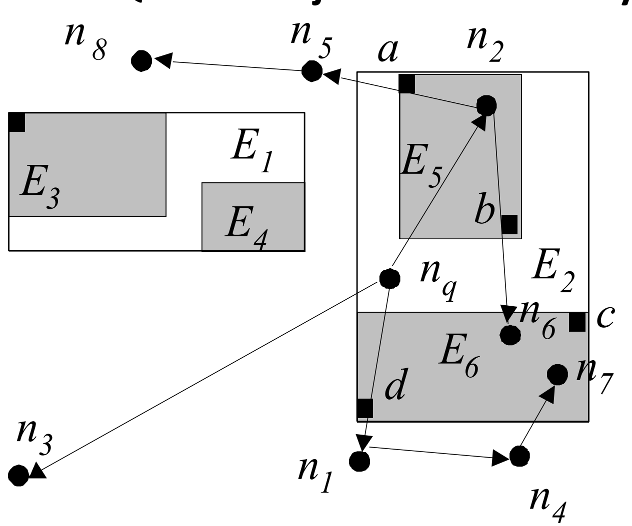
Problem? Works well when S' gets empty first
(i.e., when too many qualified segments, QS)

USC Viterbi School of Engineering

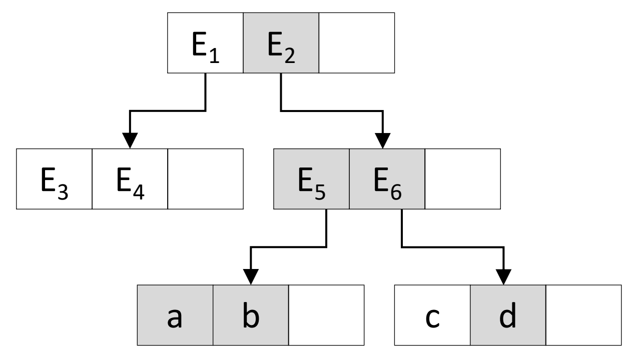
Range Queries – NE

- <u>Range Network Expansion (RNE) algorithm</u>
  - First compute the set *QS* of qualifying segments within network range *e* from *q.*
  - Retrieve the data entities falling on these segments (intersection-join between QS and objects)

QS with Object R-tree overlay



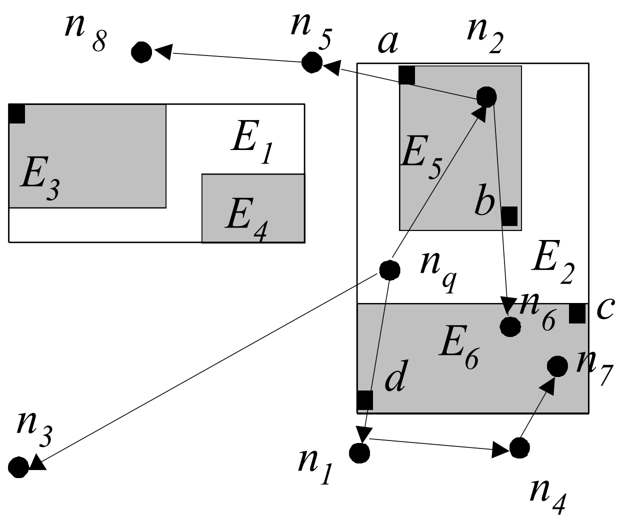
Object R-tree



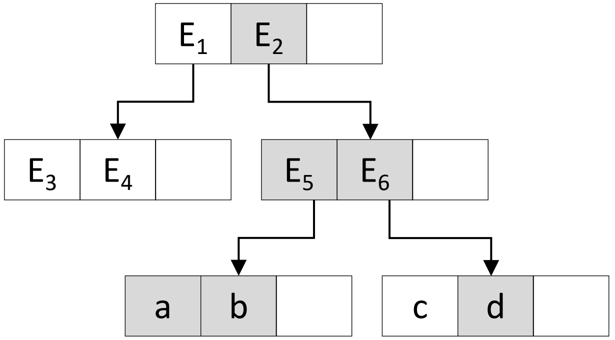
Problem? QS may be large

Range Queries – NE - Optimized

Network and Objects



Object R-tree



- Start traversing the object R-tree from root
- Compute QS_i for each entry E_i in the current R-tree node.
  - Ex: QS_1={}, QS_2={all segments except n_5n_8 and n_1n_4}, QS_5={n_qn_2, n_2n_5, n_2n_6}
  - For each entry E_i

    If $QS_i \neq$ {}, recurse down the tree
- If current node is a leaf (suppose E_6), its points only be checked against QS_6.
  - Return entities falling on the segments in QS_i (QS_6). Note that c is not a qualifying object.

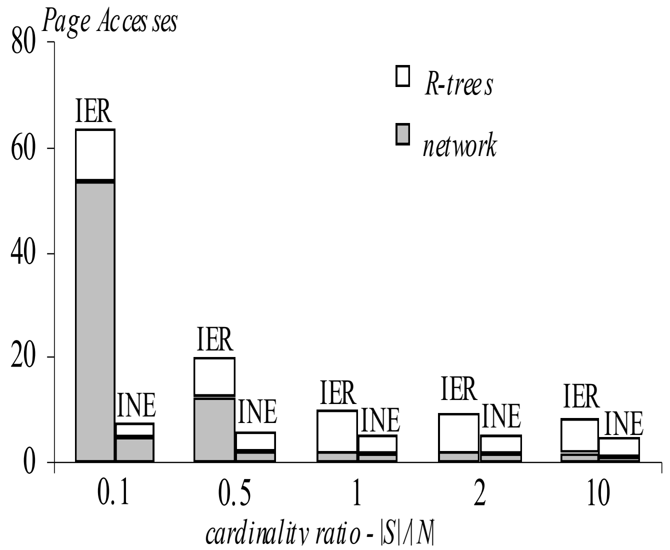
Outline

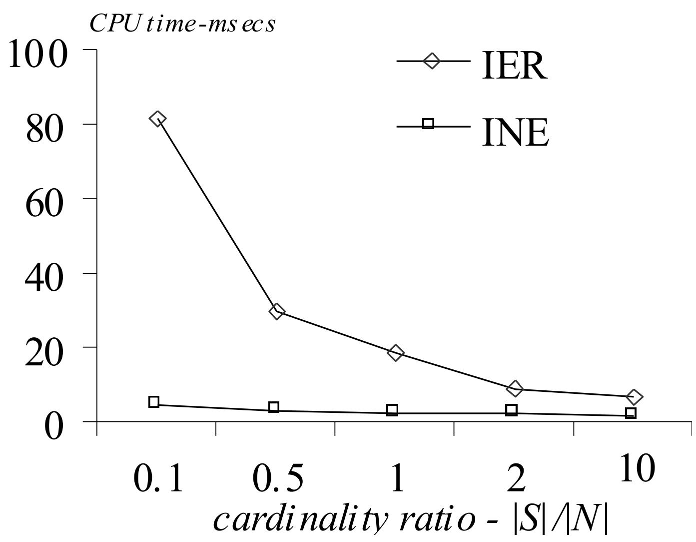
- Introduction
- Related work
  - Disk-based graph representations: 2DMatrix, CCAM structure
  - R-tree
- Spatial query in network databases
  - Architecture
  - Spatial queries:
    - Nearest neighbor query
      - Incremental Euclidean Restriction (IER) method
      - Incremental Network Expansion (INE) method
    - Range query
      - Range Euclidean Restriction (RER) method
      - Range Network Expansion (RNE) method
- Experiments
- Summary

USC Viterbi School of Engineering

Experiments - NN queries

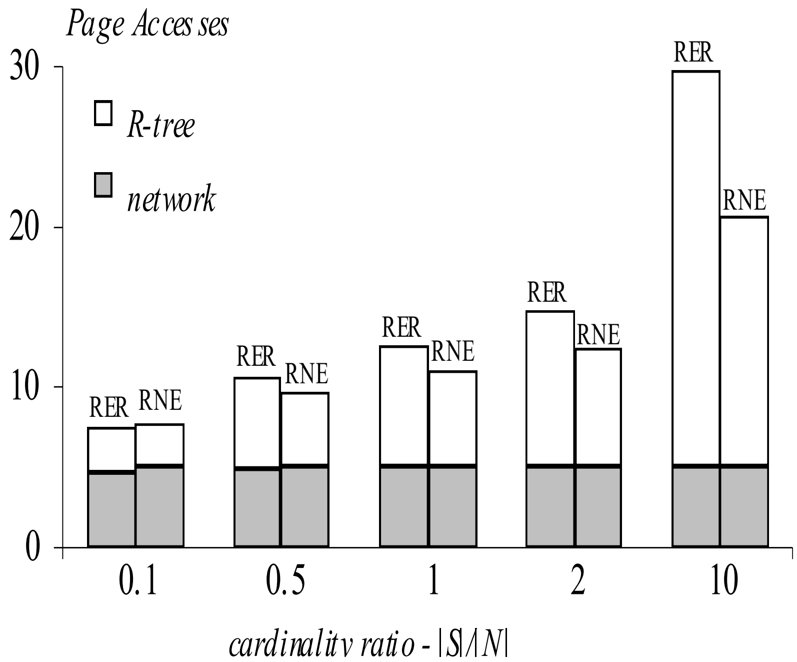
- IER (Incremental Euclidean Restriction) vs. INE (Incremental Network Expansion).
- Cost as a function of the ratio entity/edge cardinality
- Number of neighbours to be retrieved *k*=10

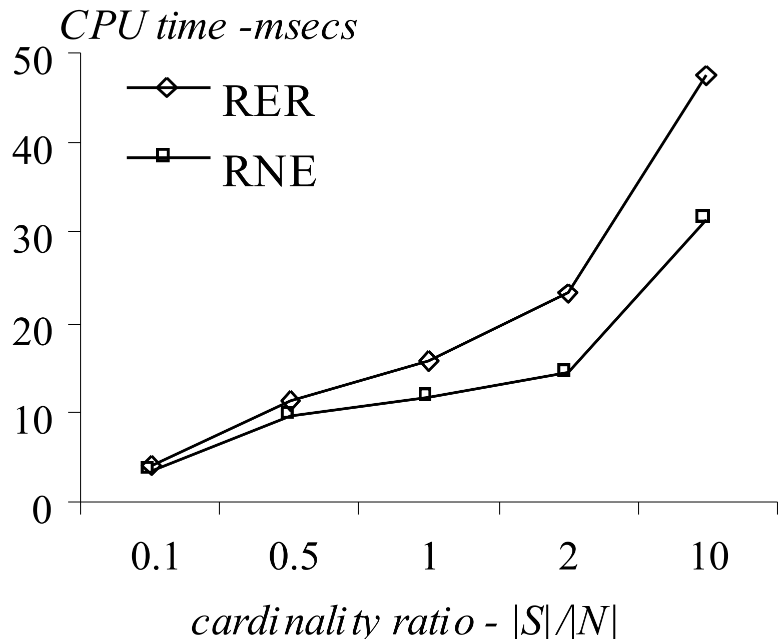




Experiments - Range Search

- RER (Range Euclidean Restriction) vs. RNE (Range Network Expansion).
- Cost as a function of the ratio entity/edge cardinality
- Length of the range *e*=1% of the data universe side length





USC Viterbi School of Engineering

Outline

- Introduction
- Related work
  - Disk-based graph representations: 2DMatrix, CCAM structure
  - R-tree
- Spatial query in network databases
  - Architecture
  - Spatial queries:
    - Nearest neighbor query
      - Incremental Euclidean Restriction (IER) method
      - Incremental Network Expansion (INE) method
    - Range query
      - Range Euclidean Restriction (RER) method
      - Range Network Expansion (RNE) method
- Experiments
- **Summary**

Summary

- Network distance is a more realistic metric than Euclidean distance.
- *Euclidean restriction* assumes the lower bounding property, which may not always hold in practice (if, for instance, the edge cost is defined as the expected travel time). On the contrary, *network expansion* permits a wide variety of costs associated with the edges.
- *Network expansion* has superior performance for nearest neighbour and range queries.

USC Viterbi
School of Engineering

References

- Dimitris Papadias, Jun Zhang, Nikos Mamoulis, Yufei Tao: Query Processing in Spatial Network Databases. VLDB 2003:802-813
- A presentation by Afsin Akdogan in csci587 Fall'2010