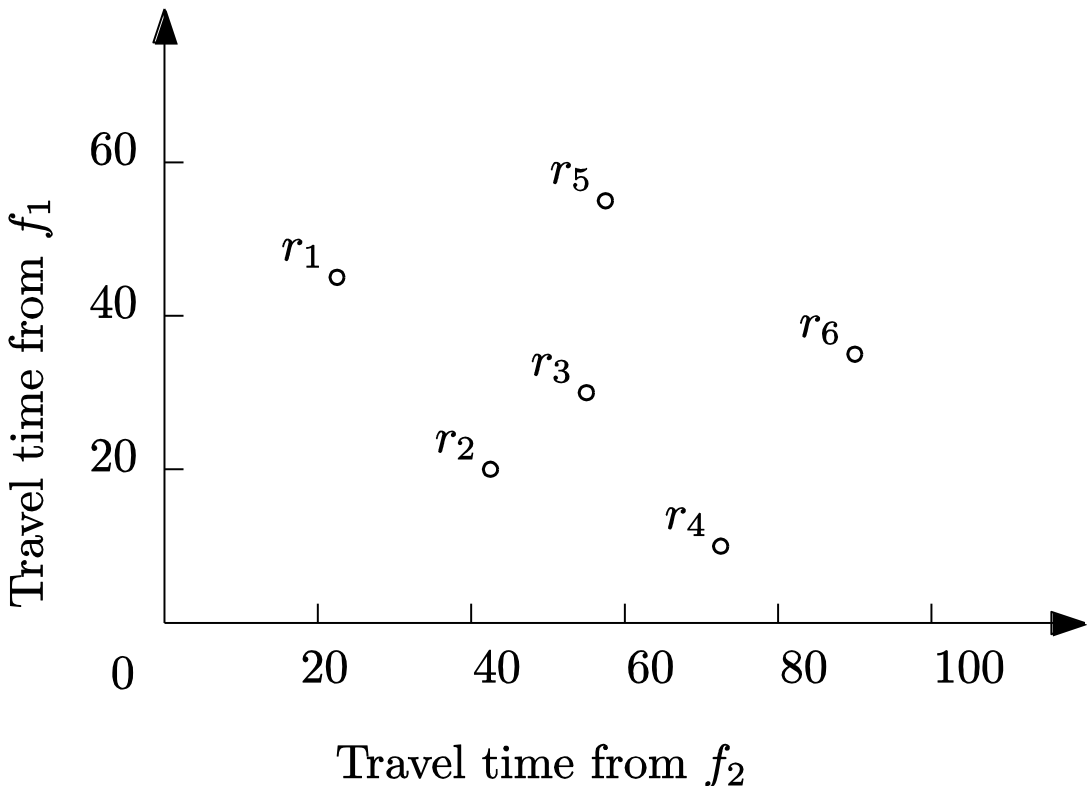
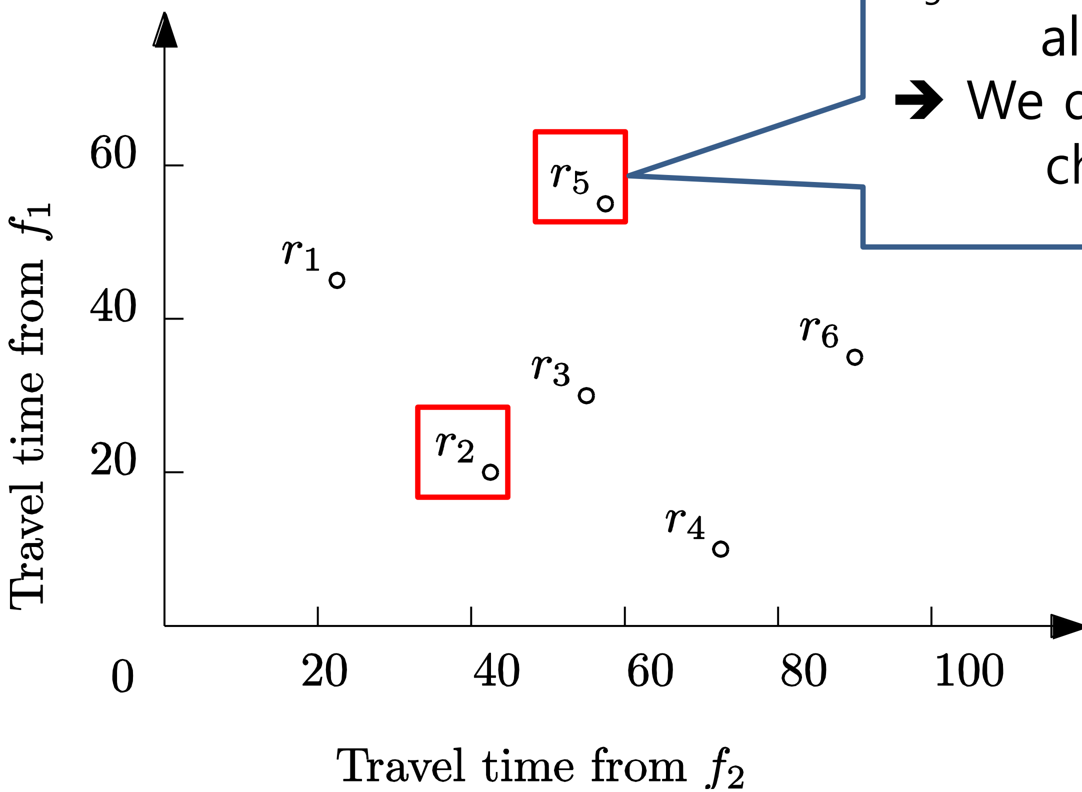
MSSQ:
Manhattan Spatial Skyline Queries

Wanbin Son
Seung-won Hwang
Hee-Kap Ahn

Introduction



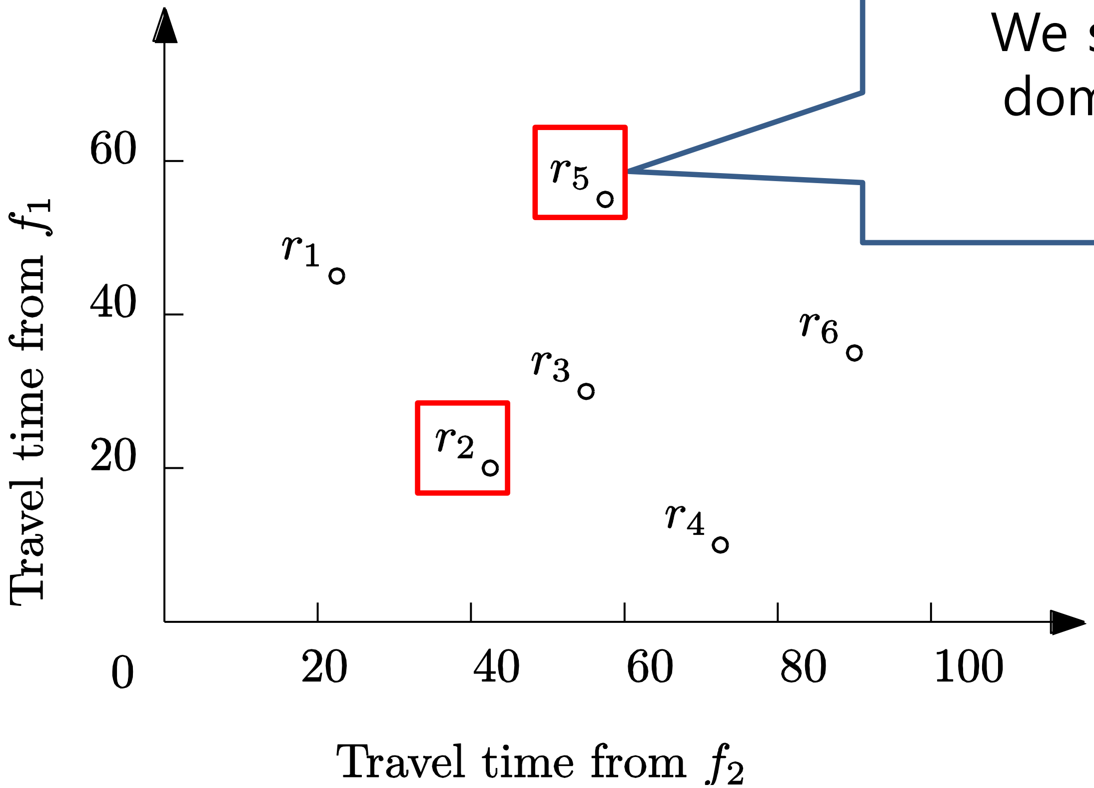
Introduction



r_5 is worse than r_2 for all factors.

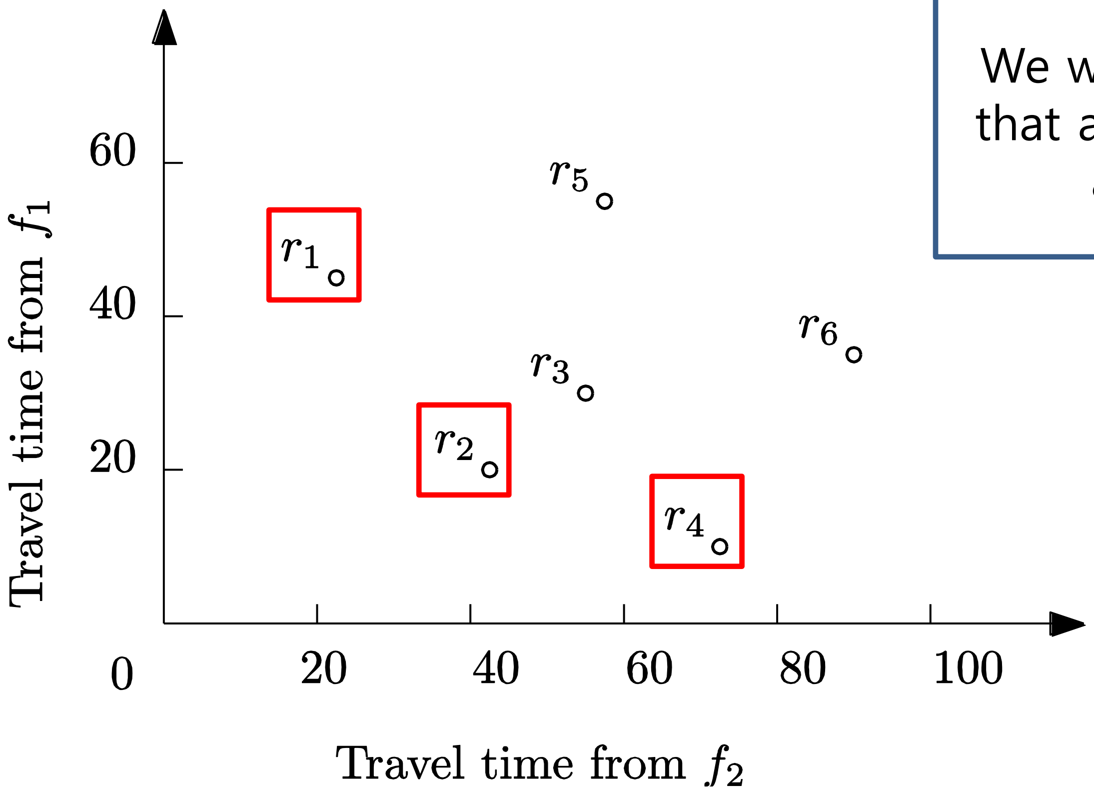
➔ We do not need to choose r_5.

Introduction



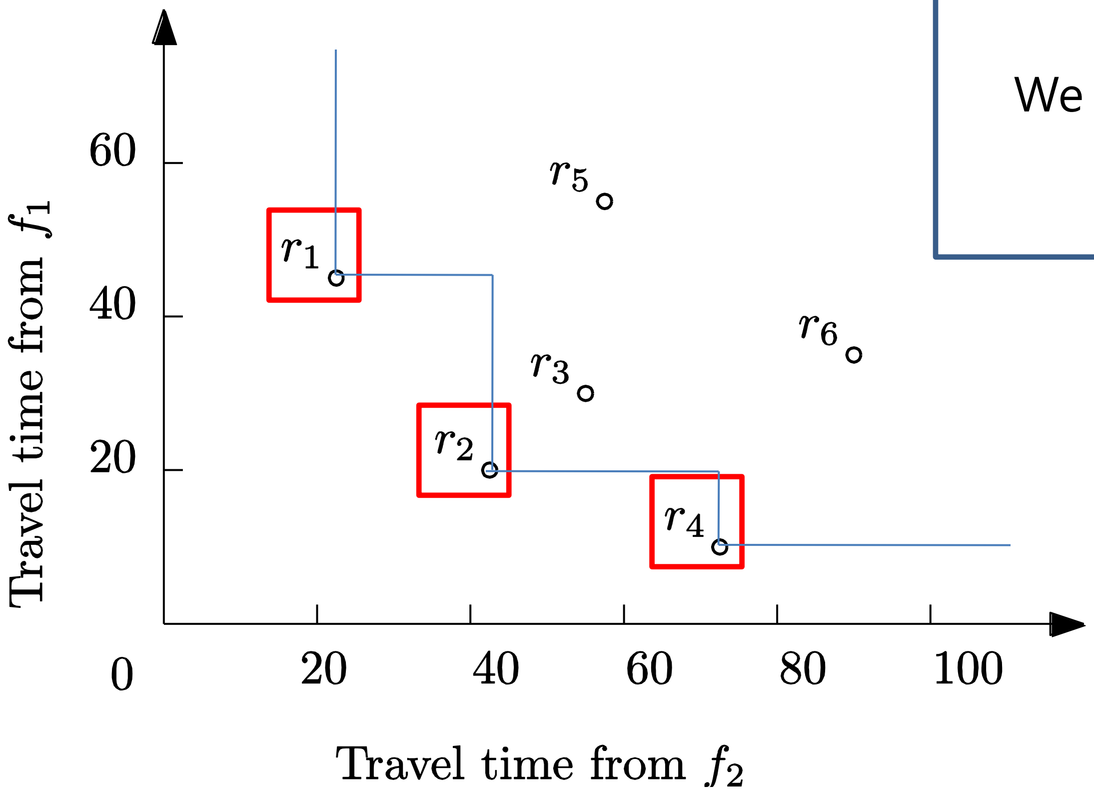
We say that r_2 dominates r_5.

Skyline



We want to find all points that are not dominated by any other point.

Skyline



We call these points to skylines.

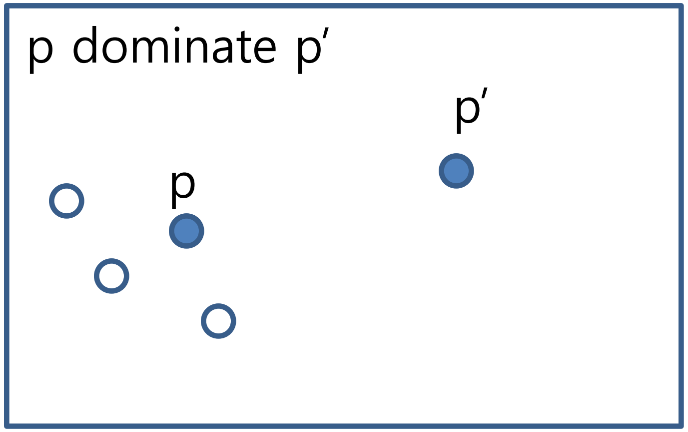
Problem Definition

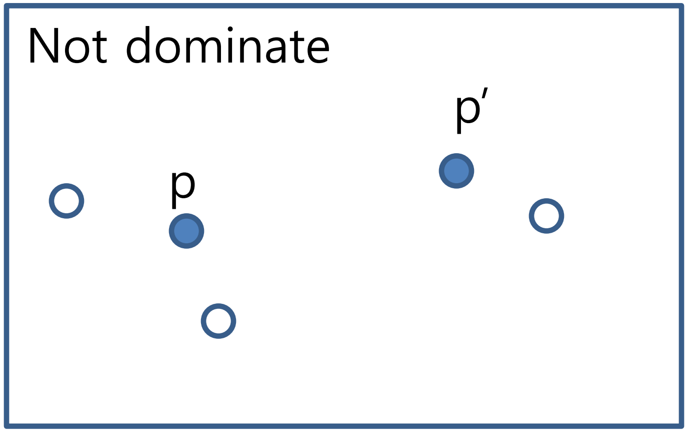
In the plane

- Input
  - P : the set of data points
  - Q : the set of query points ($|P| \geq |Q|$)
- output
  - S : the skyline ($S \subseteq P$)

We want to compute the set S of all spatial skyline points from P with respect to Q in the plane

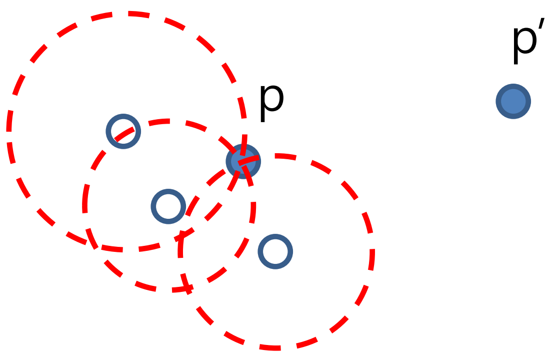
Problem Definition

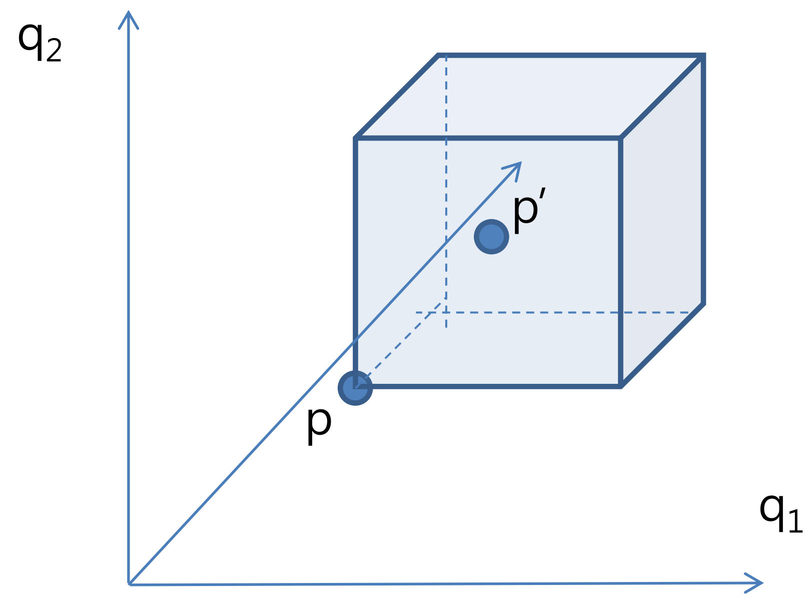




Problem Definition

p dominate p′



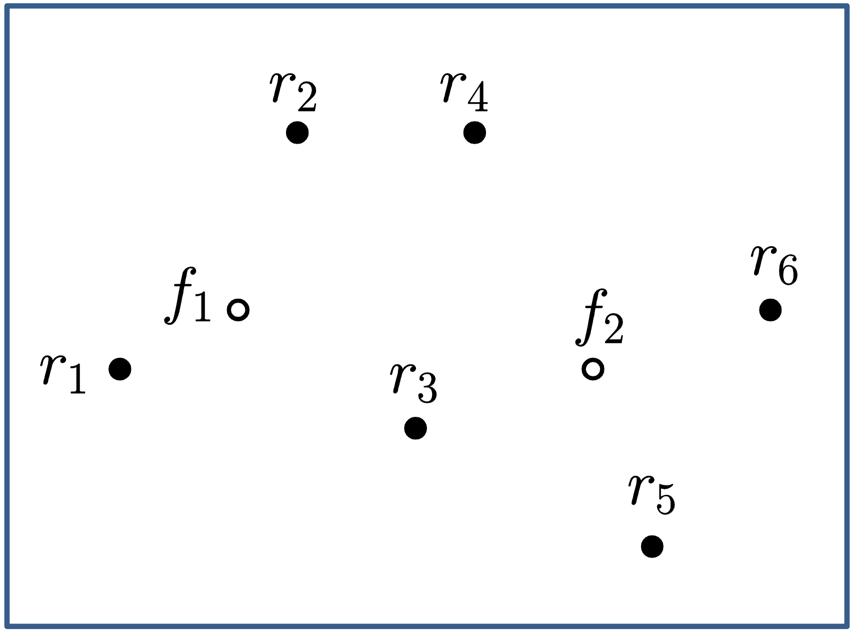


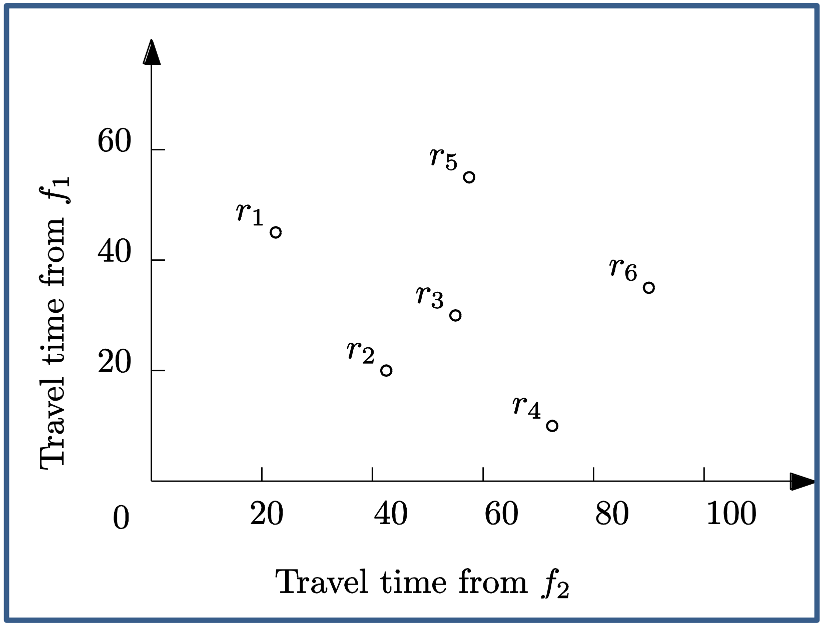
- **Definition : dominate**

  We say that p_1 dominates p_2 if and only if $d(p_1,q) \leq d(p_2,q)$ for every $q \in Q$ and $d(p_1,q') < d(p_2,q')$ for some $q' \in Q$.

Problem Definition

P={r_1, r_2, r_3, r_4, r_5,} Q={f_1, f_2}





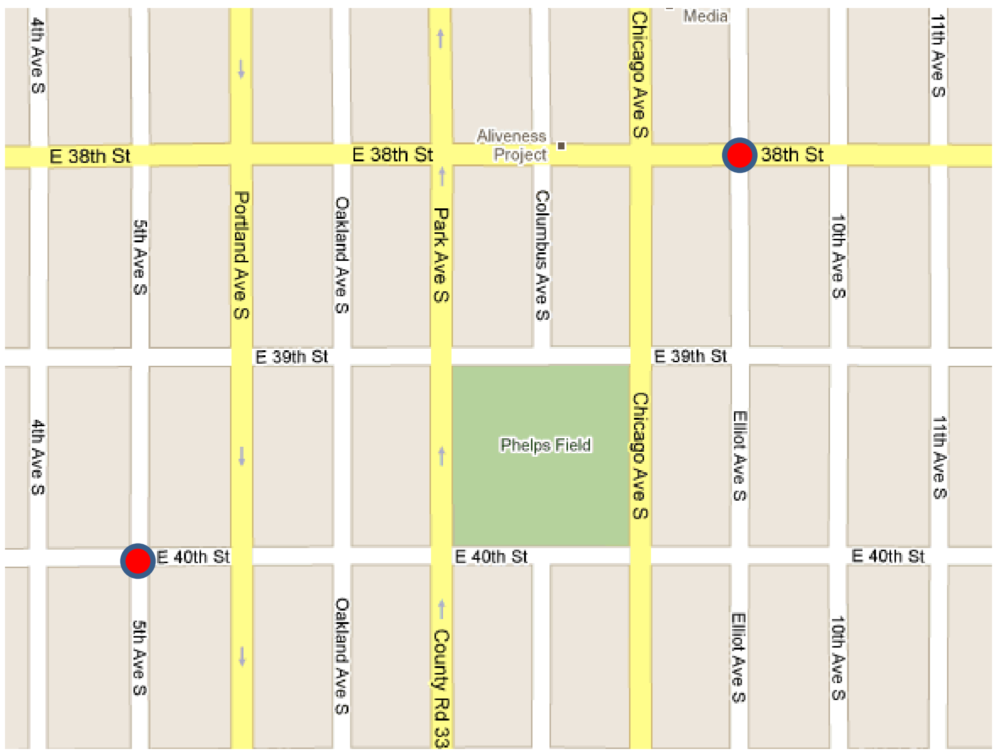
L_1 metric

- L_1 distance between two points (x_1,y_1) and (x_2,y_2) in the plane is $|x_1-x_2|+|y_1-y_2|$.

- L_1 metric is reasonable to represent a well-developed road network.

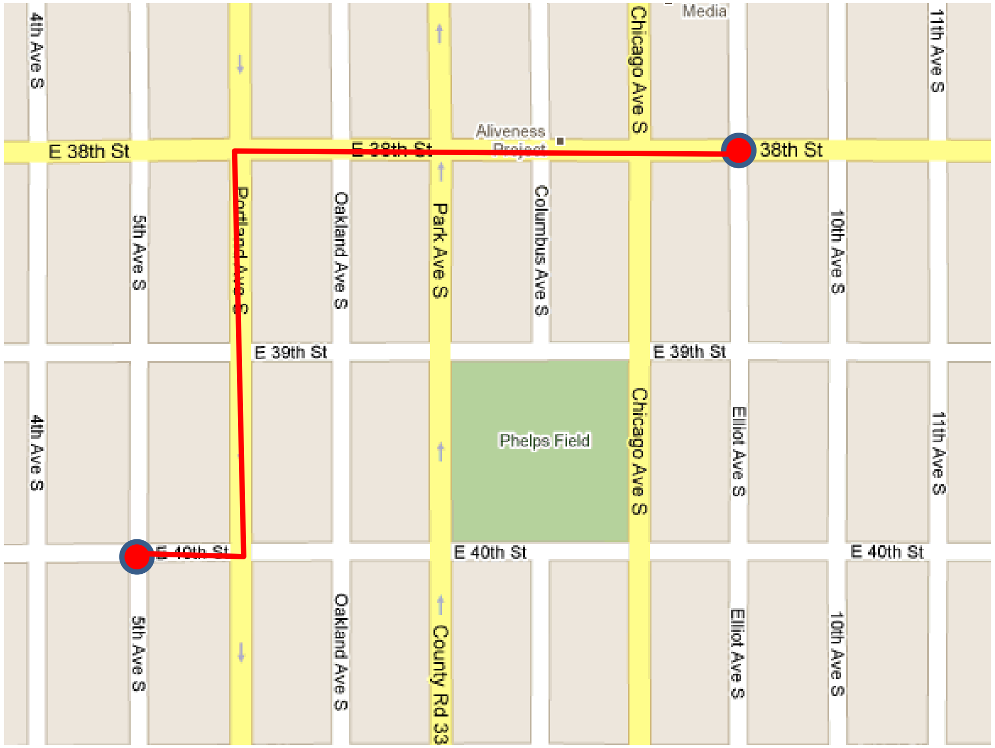
L_1 metric

- L_1 metric is reasonable to represent a well-developed road network.



L_1 metric

- L_1 metric is reasonable to represent a well-developed road network.



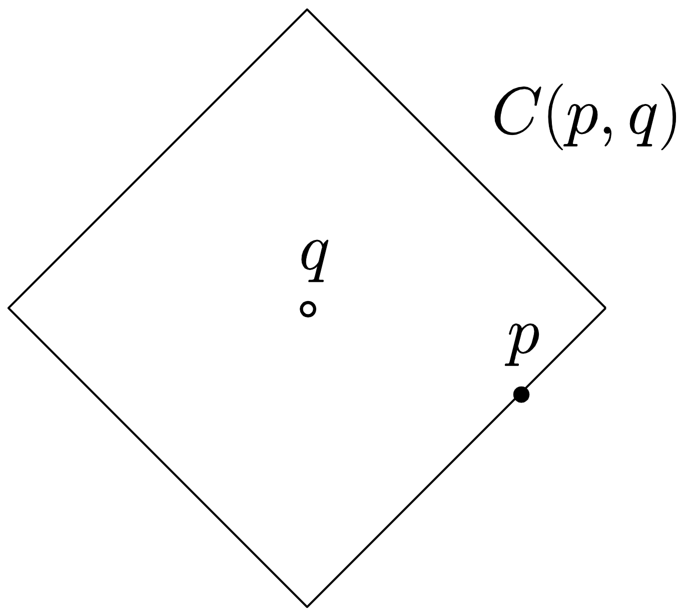
Related work

- Spatial skyline queries for the other metrics.
  - Based on pairwise dominance test

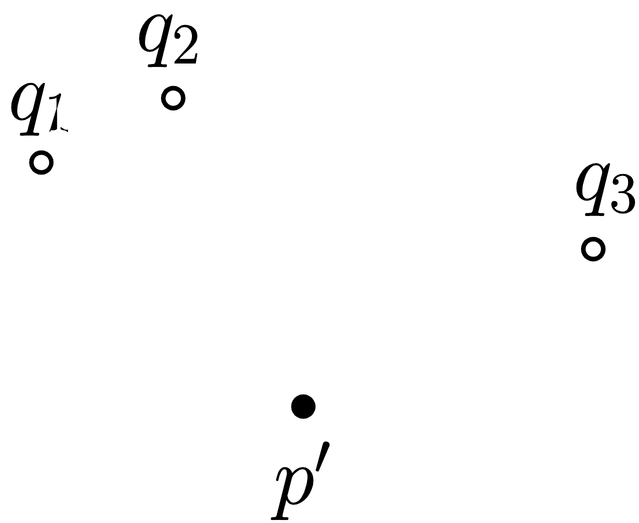
- Skyline queries
  - Based on pairwise dominance test
  - Operations for d dimensinal data

L_1 metric

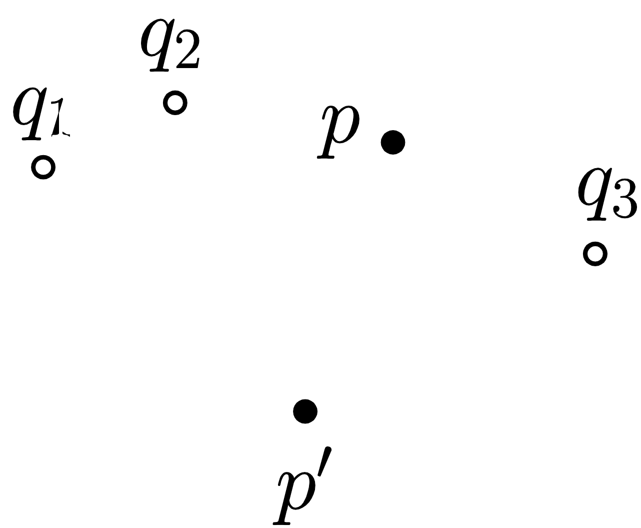
In the plane, we use L_1 metric for the distance between two points.



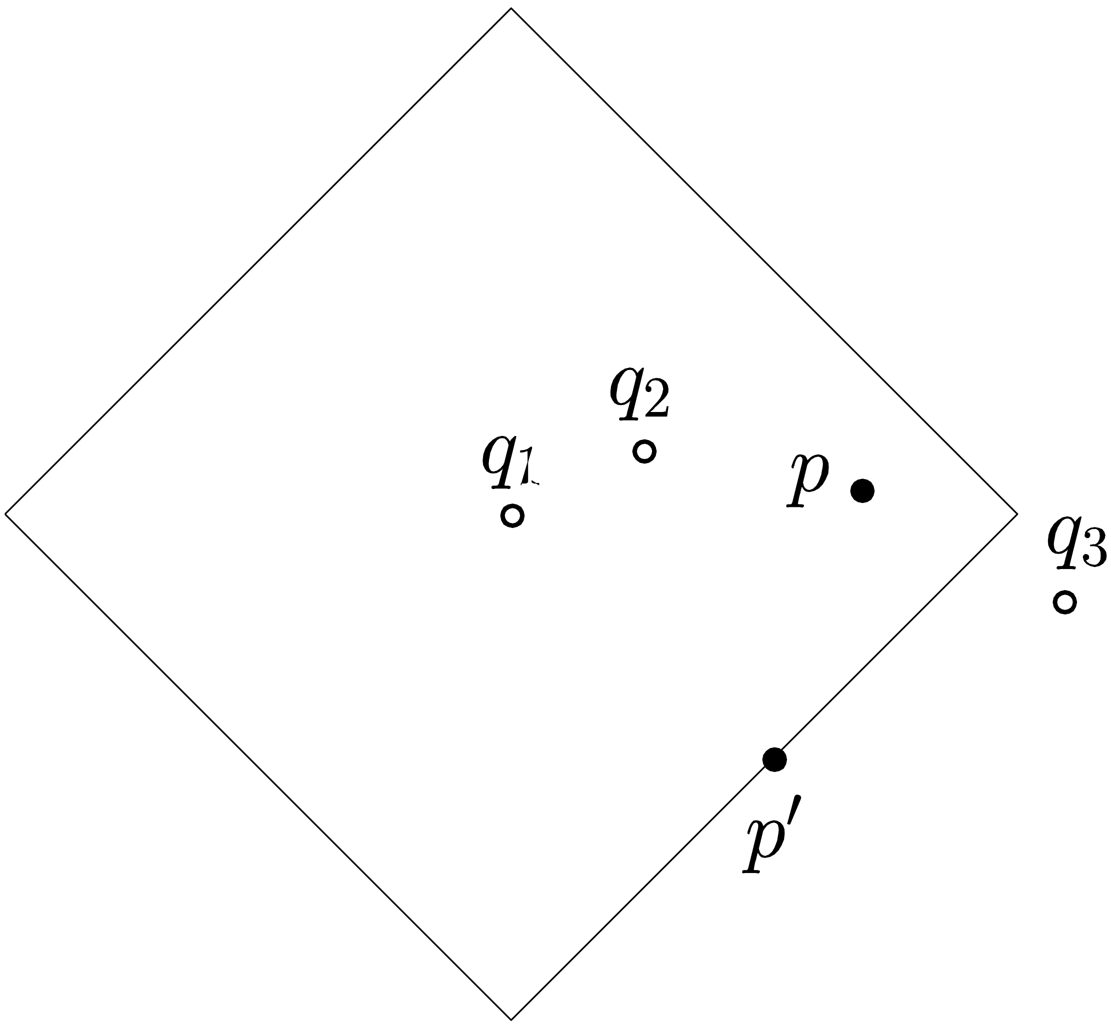
Observation



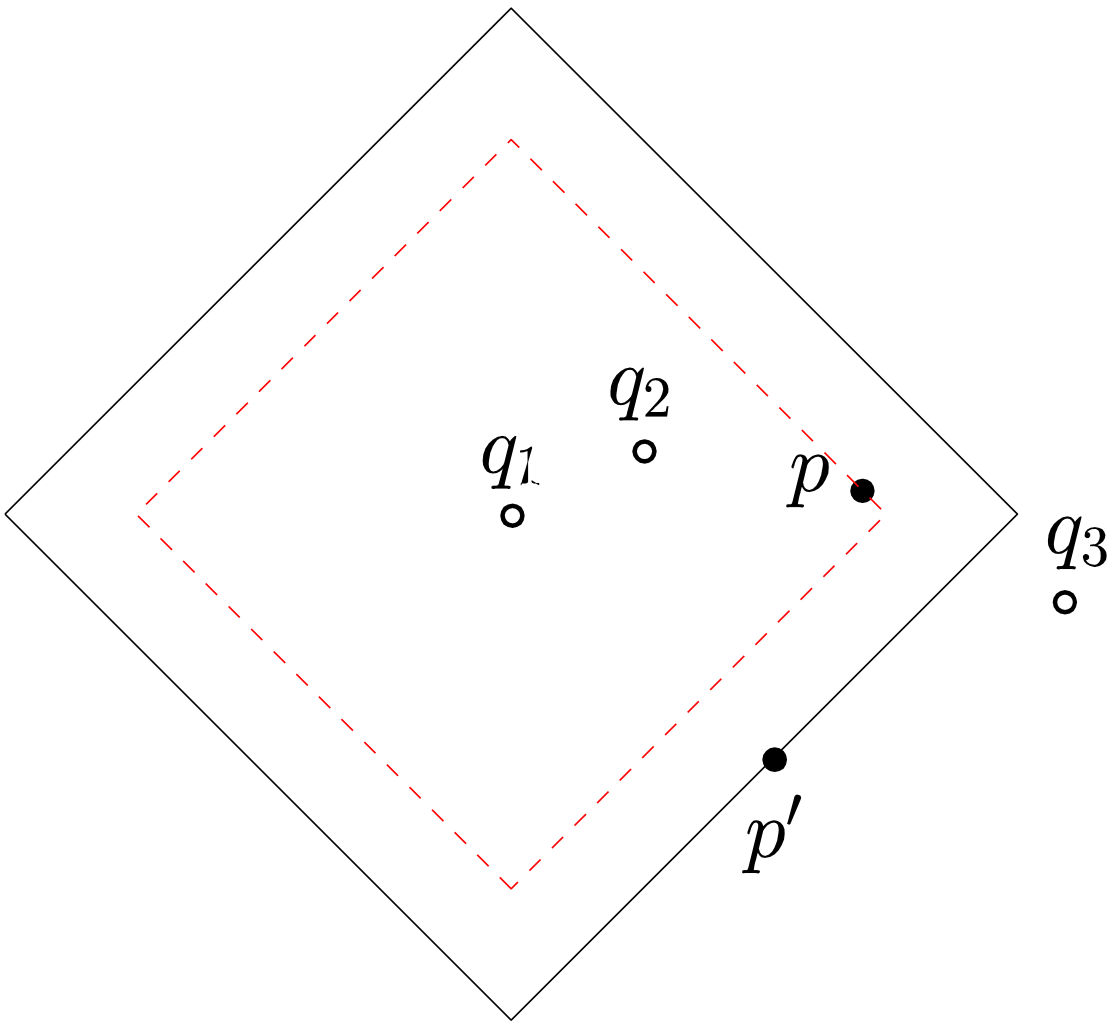
Observation



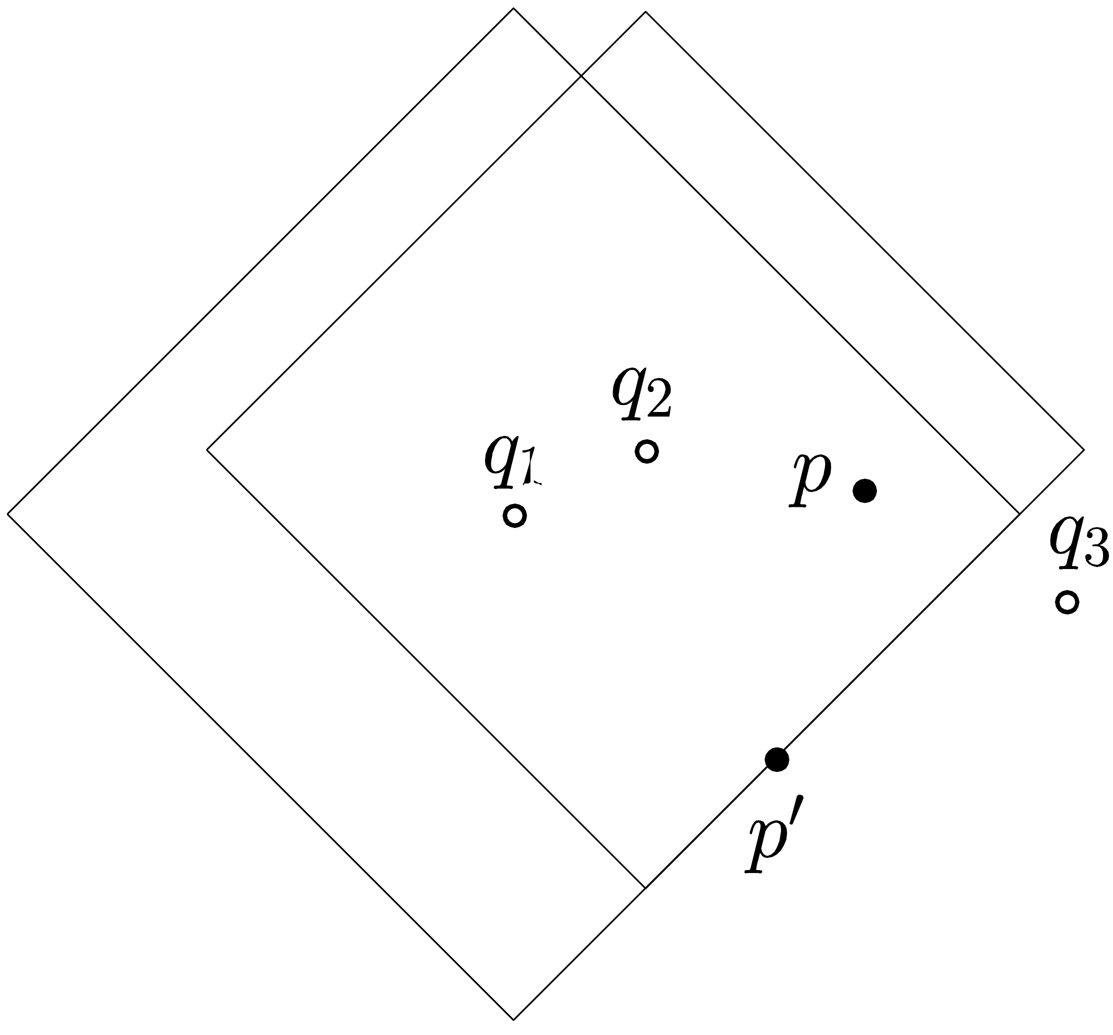
Observation



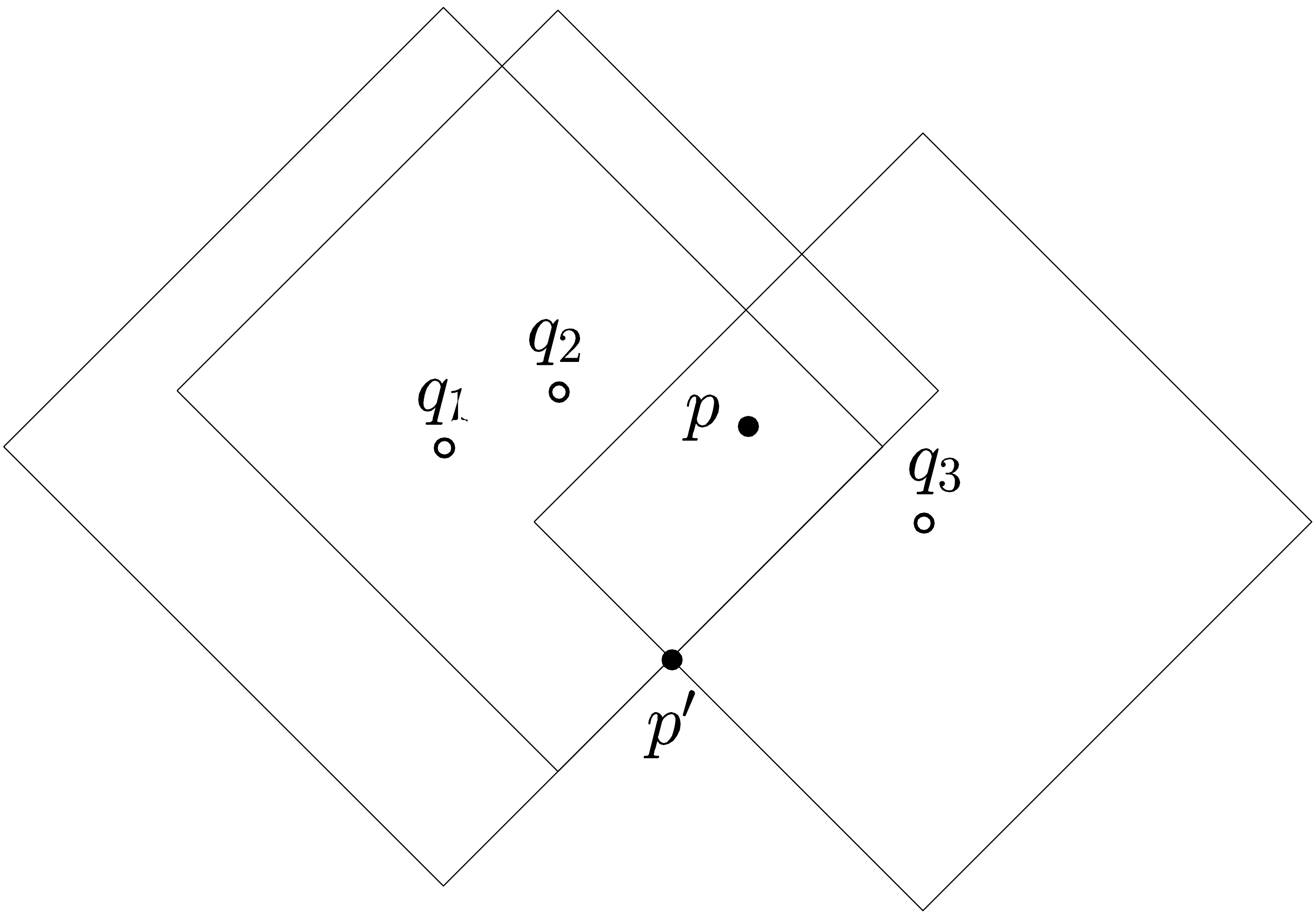
Observation



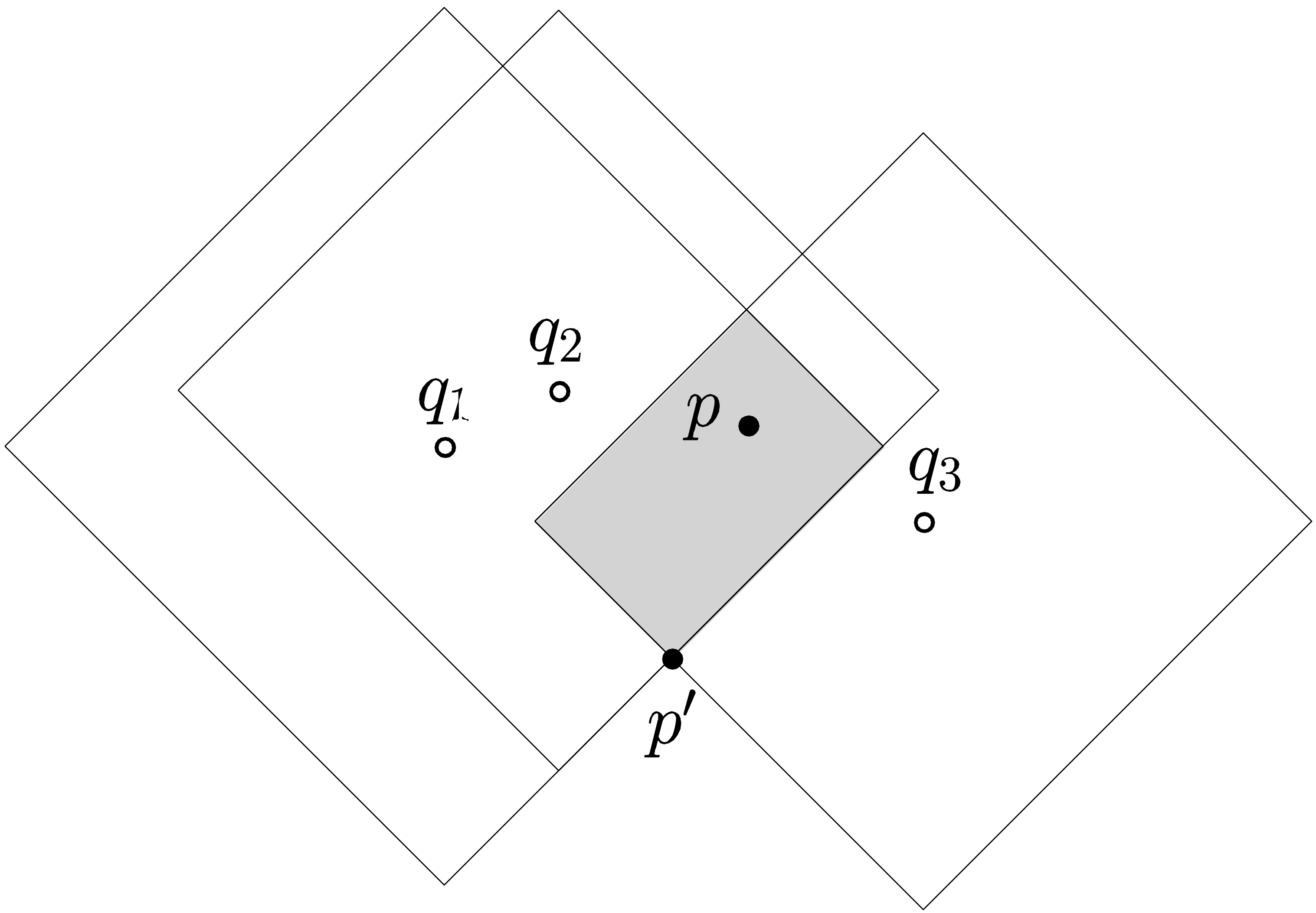
Observation



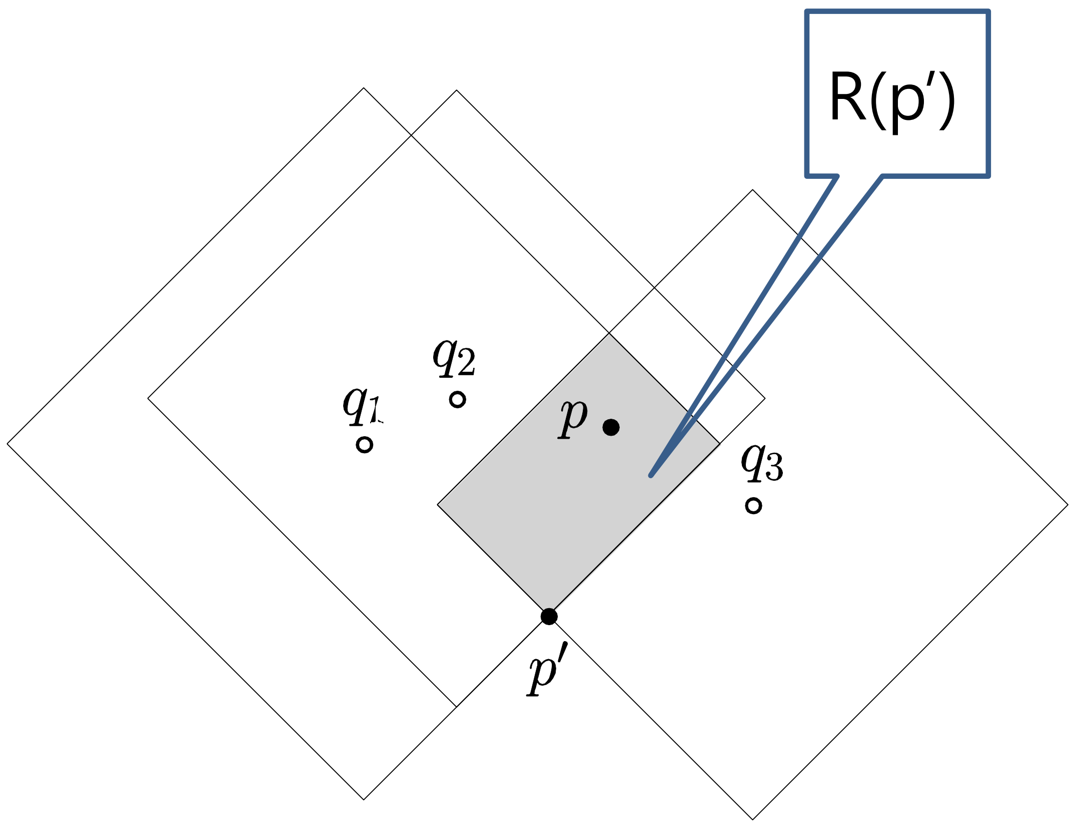
Observation



Observation



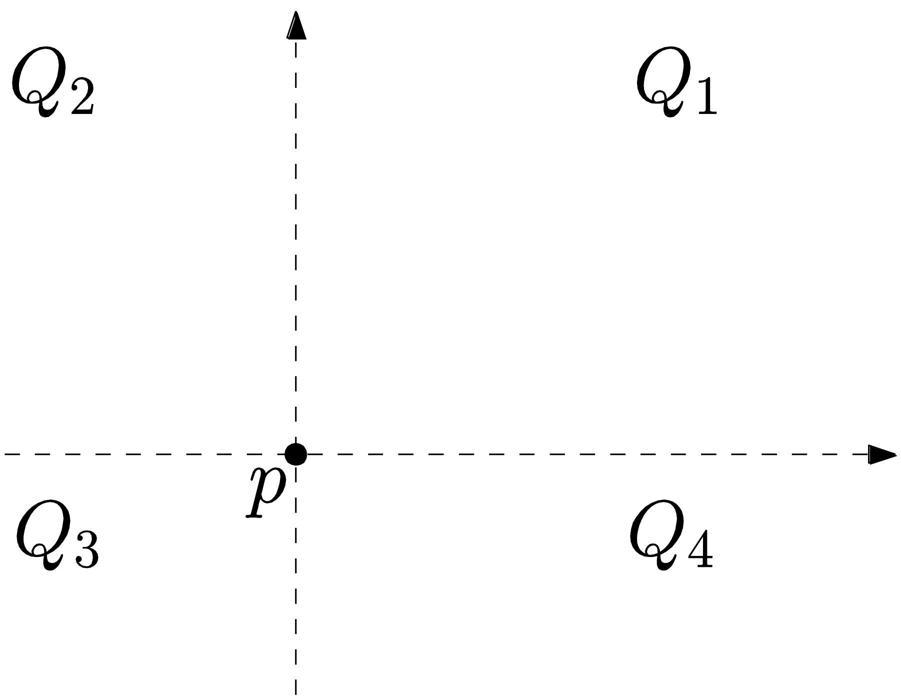
Observation



Computing R(p′) takes O(|Q|) time for each point in P

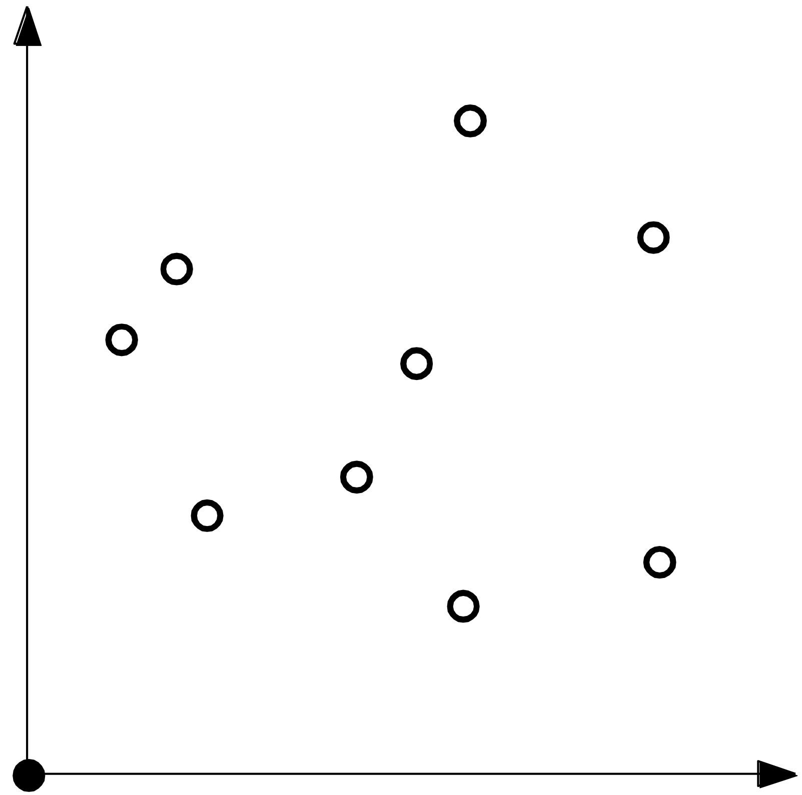
Computing R(p)

- There are at most four query points that determine the R(p)

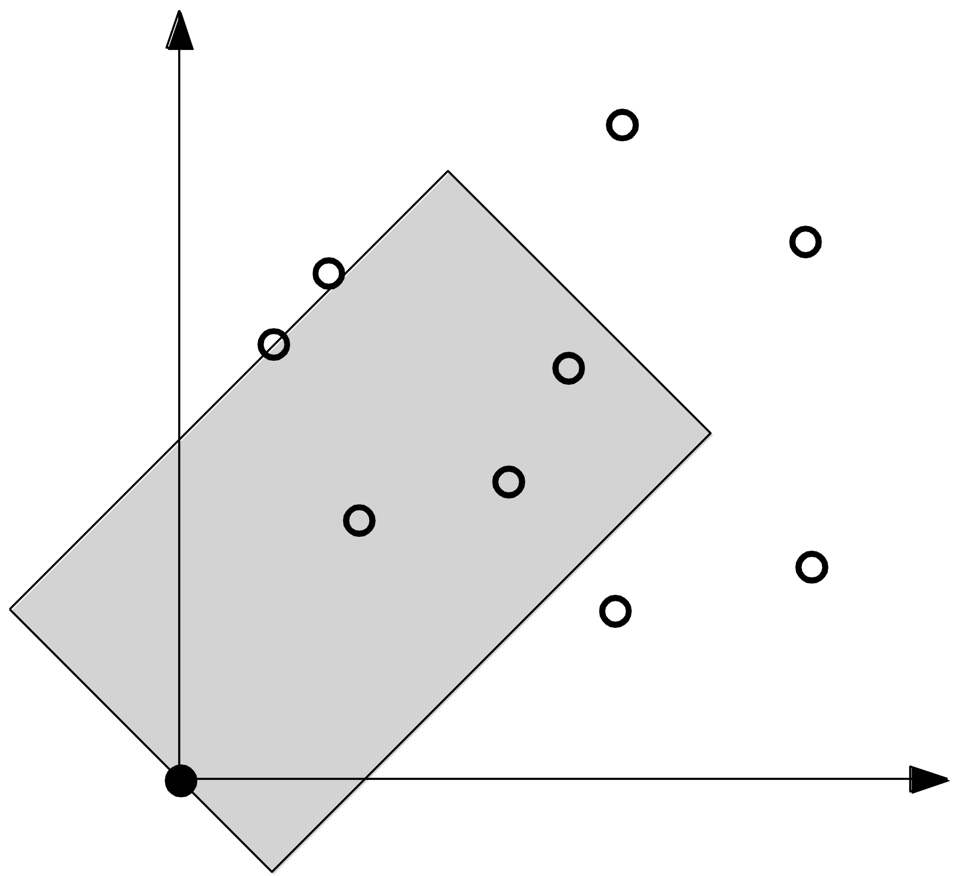


Computing R(p)

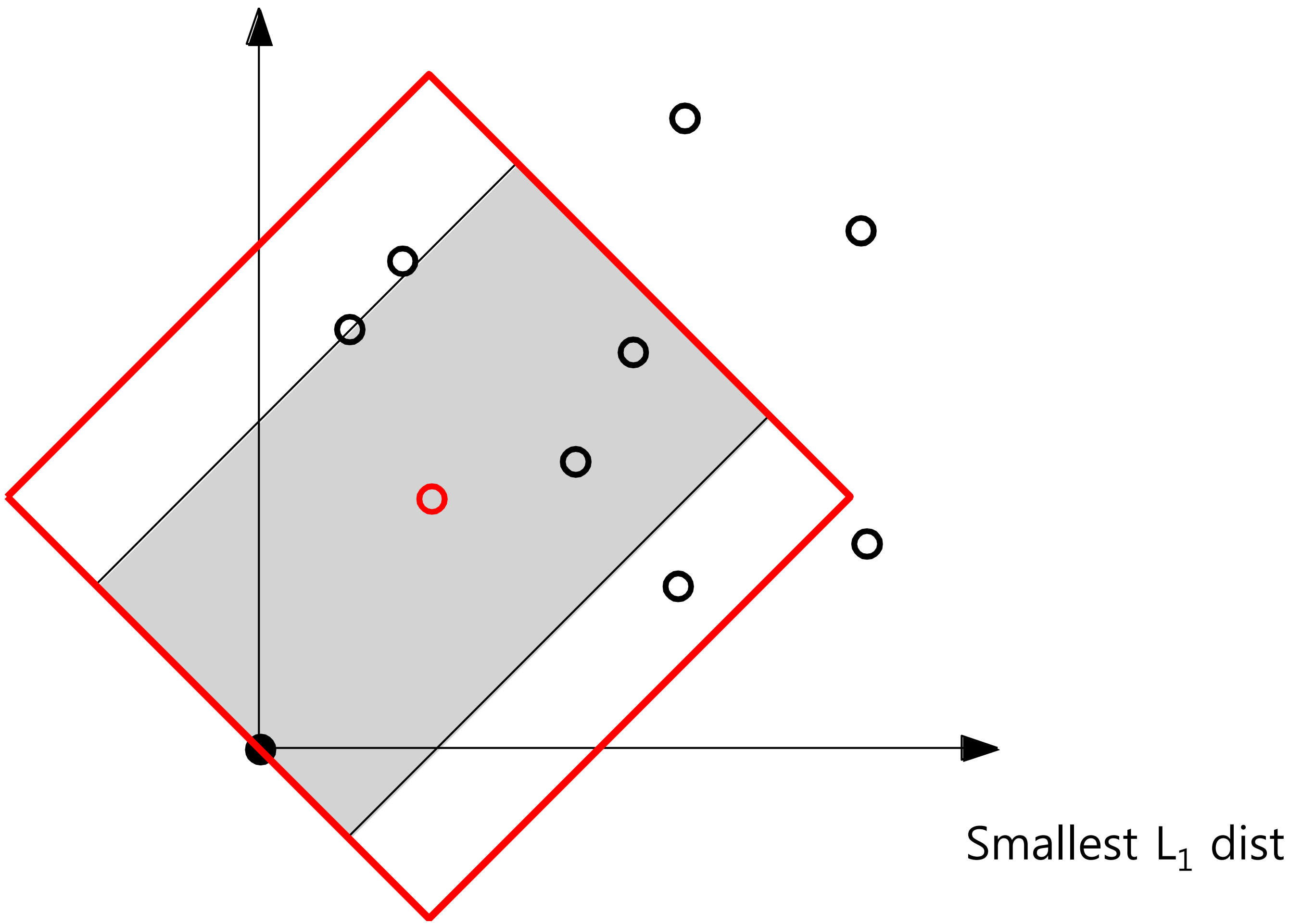
$Q_1 = Q$



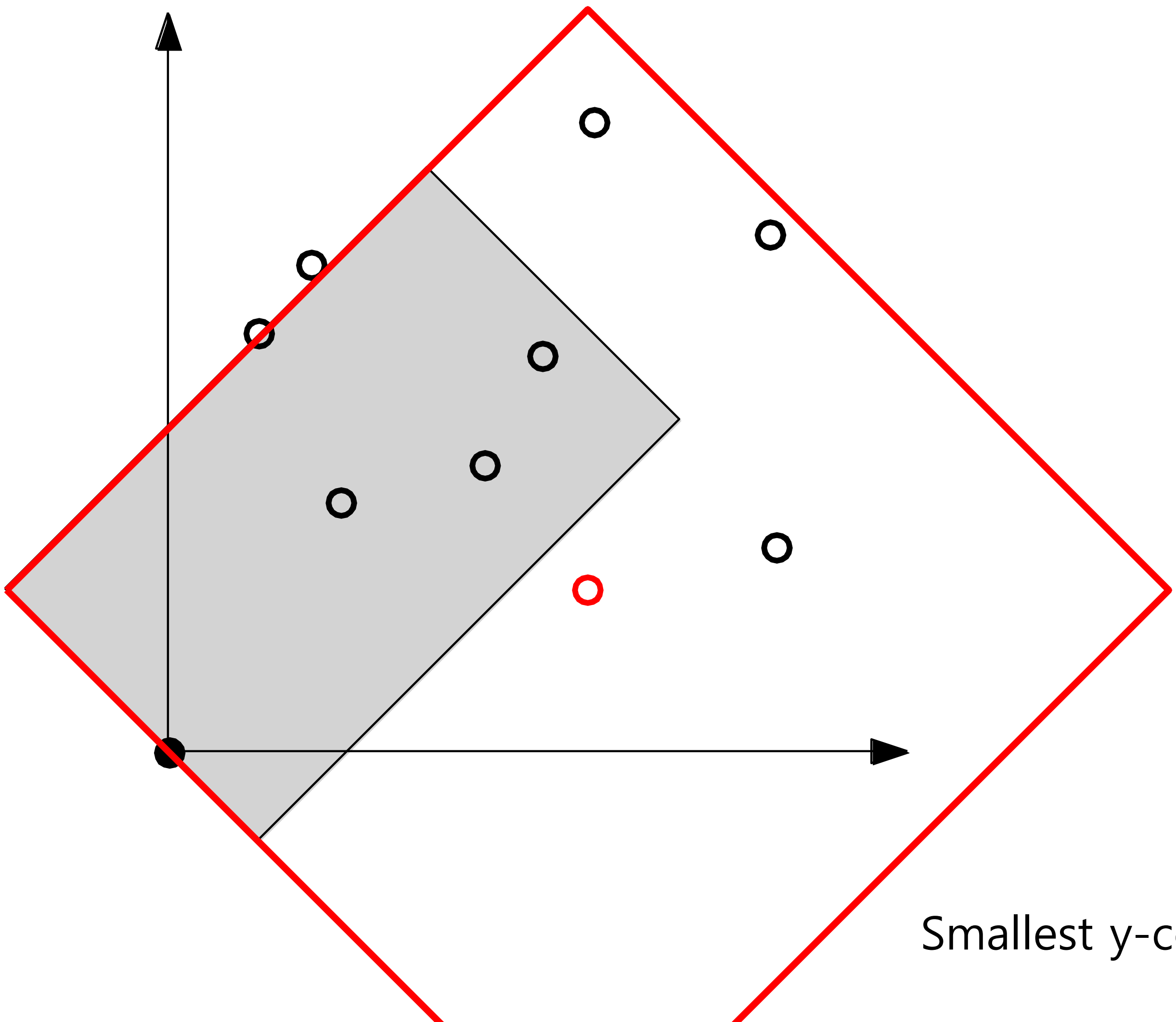
Computing R(p)



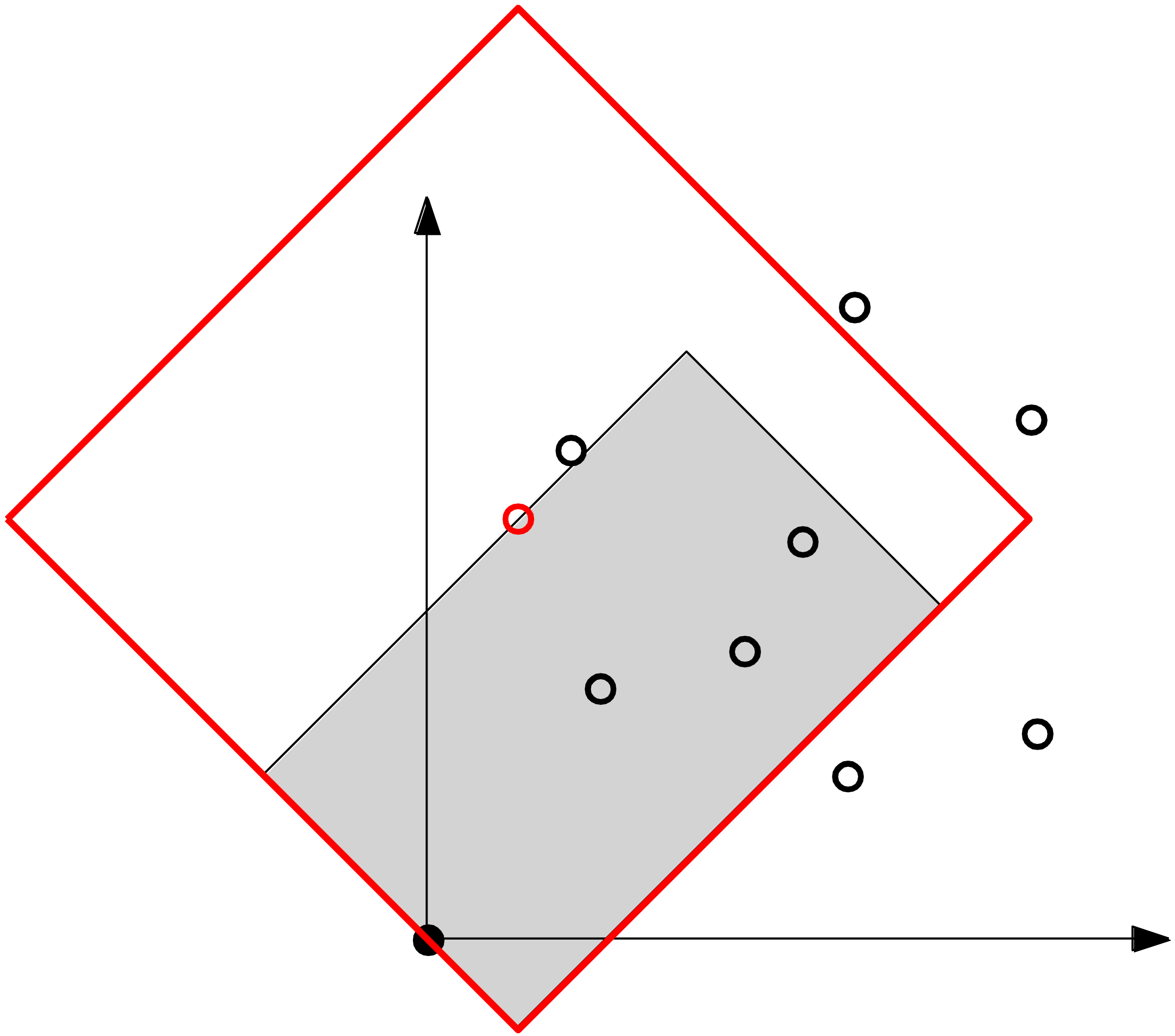
Computing R(p)



Computing R(p)



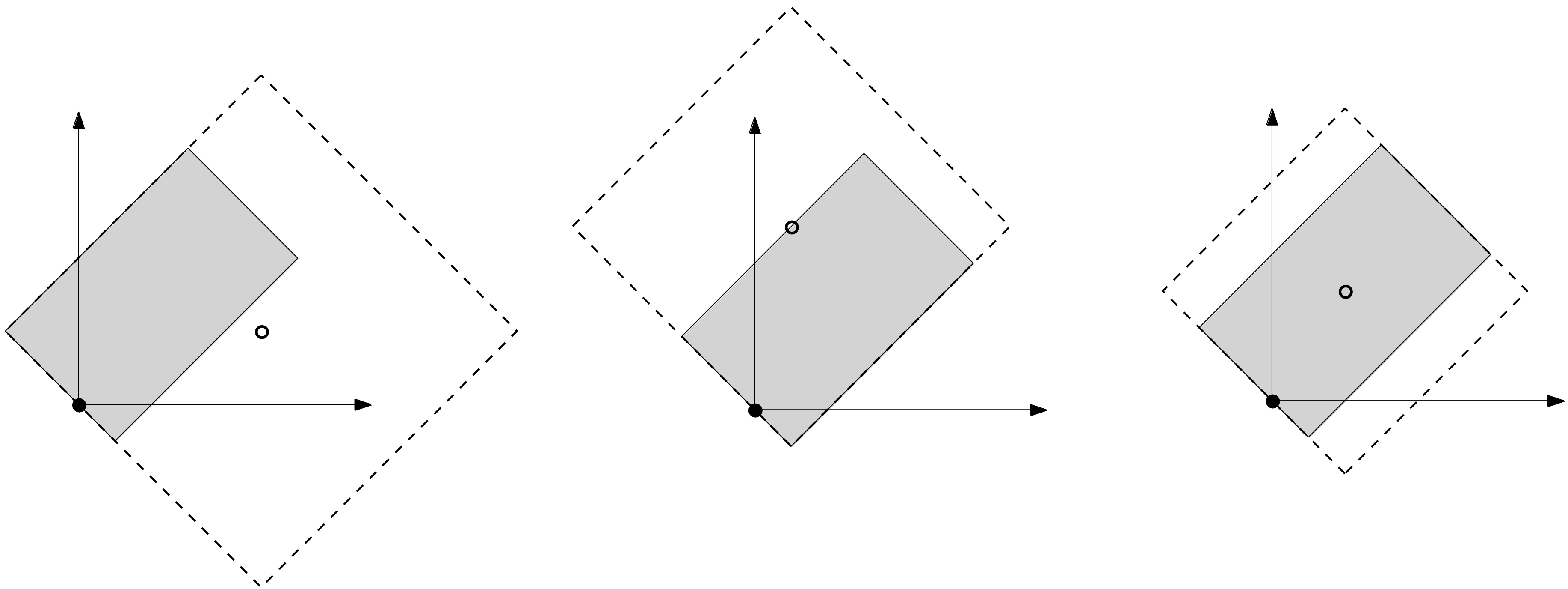
Smallest y-coordinate



Smallest x-coordinate

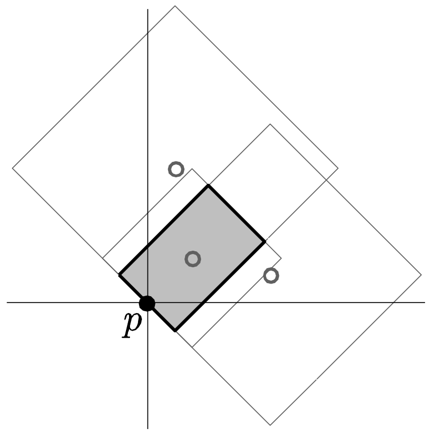
Computing R(p)

The case that $Q_1=Q$

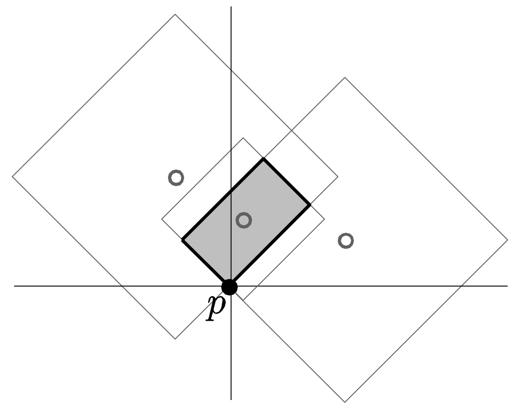


Smallest y-coordinate   Smallest x-coordinate   Smallest L_1 dist

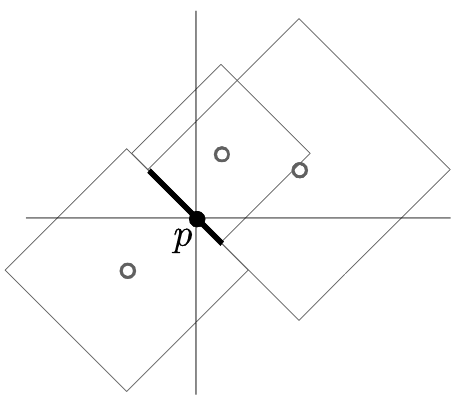
Computing R(p)



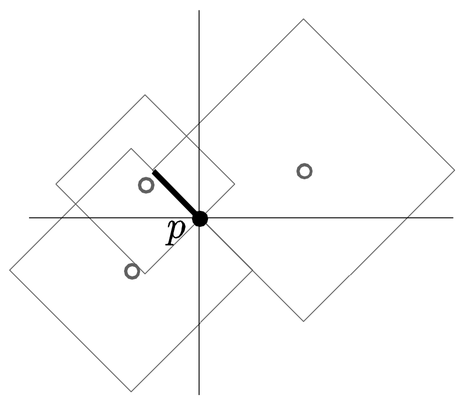
$Q_1 = Q$



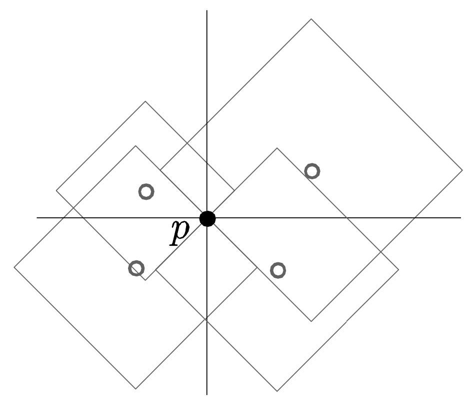
$Q_1 \cup Q_2 = Q$



$Q_1 \cup Q_3 = Q$



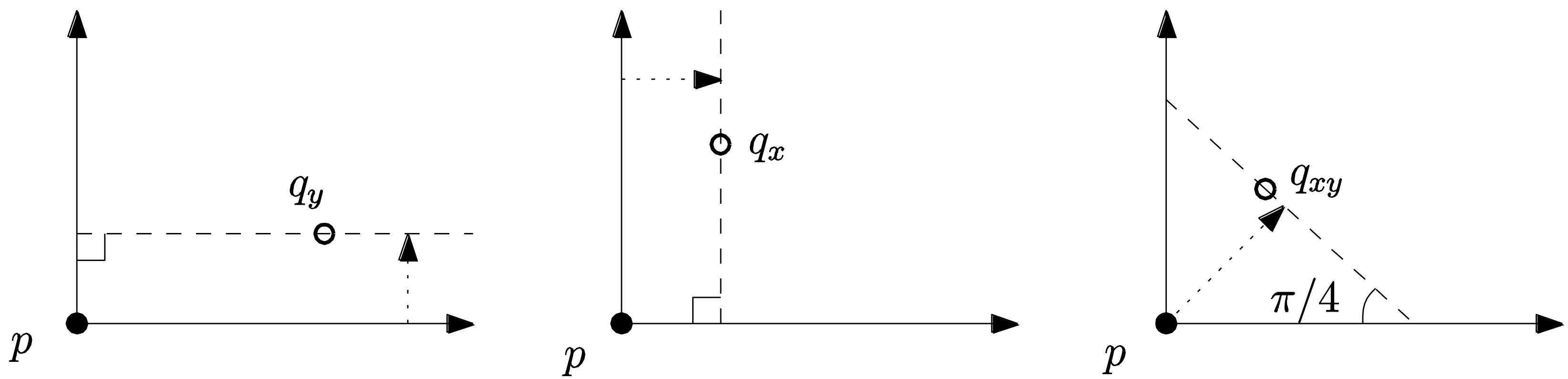
$Q_1 \cup Q_2 \cup Q_3 = Q$



$Q_1 \cup Q_2 \cup Q_3 \cup Q_4 = Q$

Computing R(p)

Using segment dragging queries



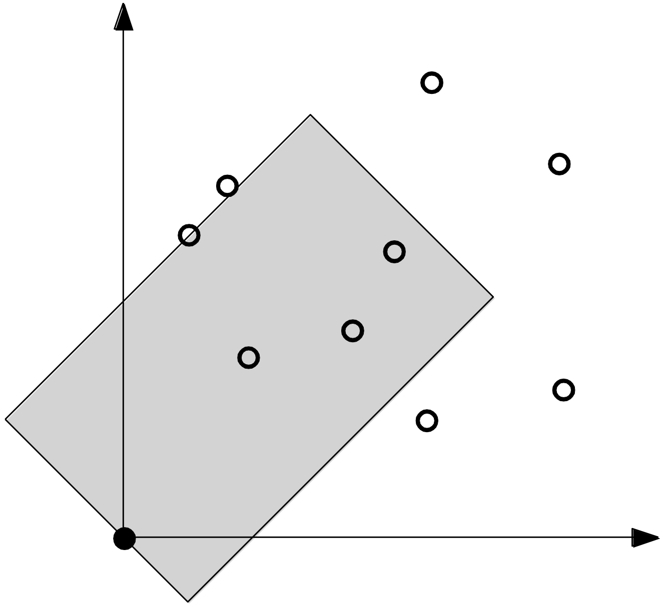
O(n log n) time preprocessing

O(log n) time for each query

O(n) space

Range counting query

- Using range counting query to count the number of data points in R(p)



Three cases for R(p)

- Three cases for data points contained in R(p)

Three cases for R(p)

- Three cases for data points contained in R(p)

  (a) no data point in R(p), other than p

  ➔ p is skyline

Three cases for R(p)

- Three cases for data points contained in R(p)

  (a) no data point in R(p), other than p

  (b) some data point p′ in the interior of R(p)

  ➔ p is not skyline

Three cases for R(p)

- Three cases for data points contained in R(p)

  (a) no data point in R(p), other than p

  (b) some data point p′ in the interior of R(p)

  (c) some data point p′ in R(p), but no data point in the interior of R(p)

Three cases for R(p)

- Three cases for data points contained in R(p)

  (a) no data point in R(p), other than p

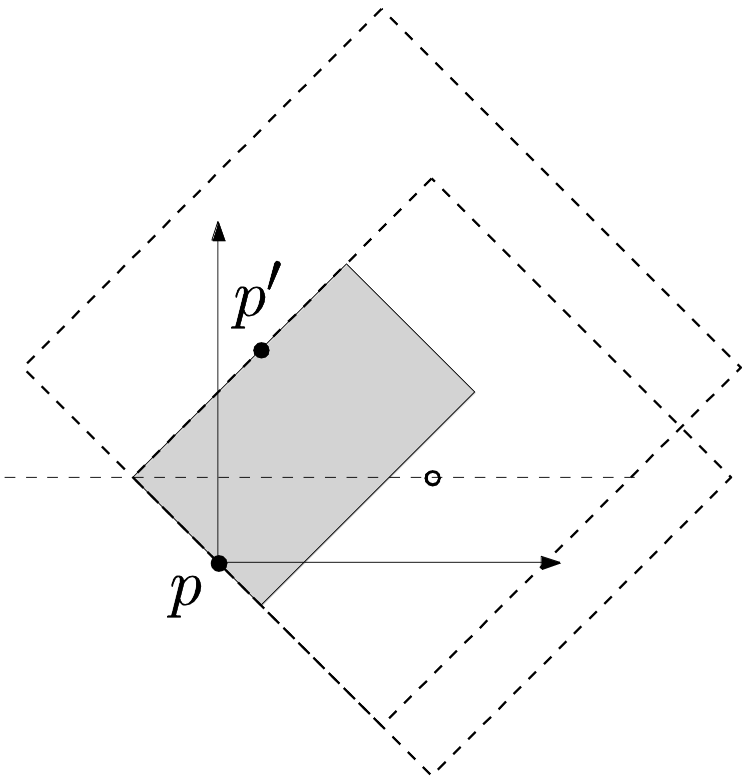
  (b) some data point p′ in the interior of R(p)

  (c) some data point p′ in R(p), but no data point in the interior of R(p)

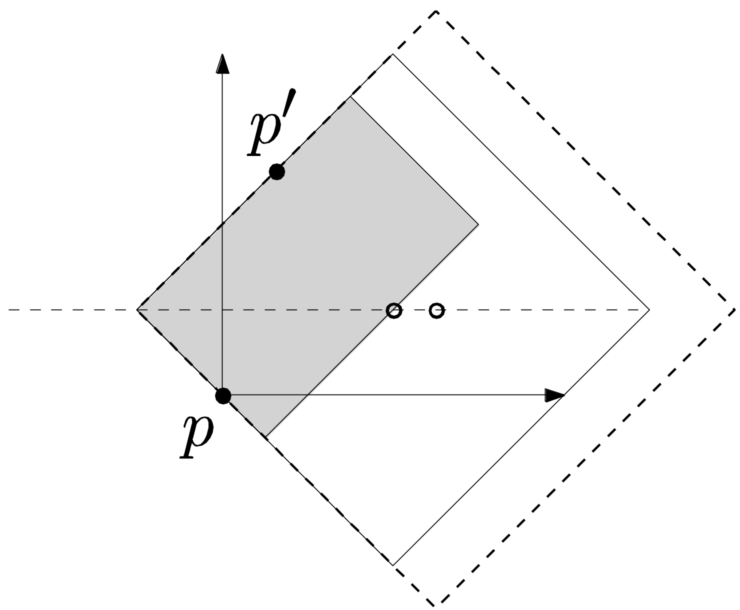
- **Definition : dominate**

  We say that p_1 dominates p_2 if and only if $d(p_1,q) \leq d(p_2,q)$ for every $q \in Q$ and $d(p_1,q') < d(p_2,q')$ for some $q' \in Q$.

Case (c)



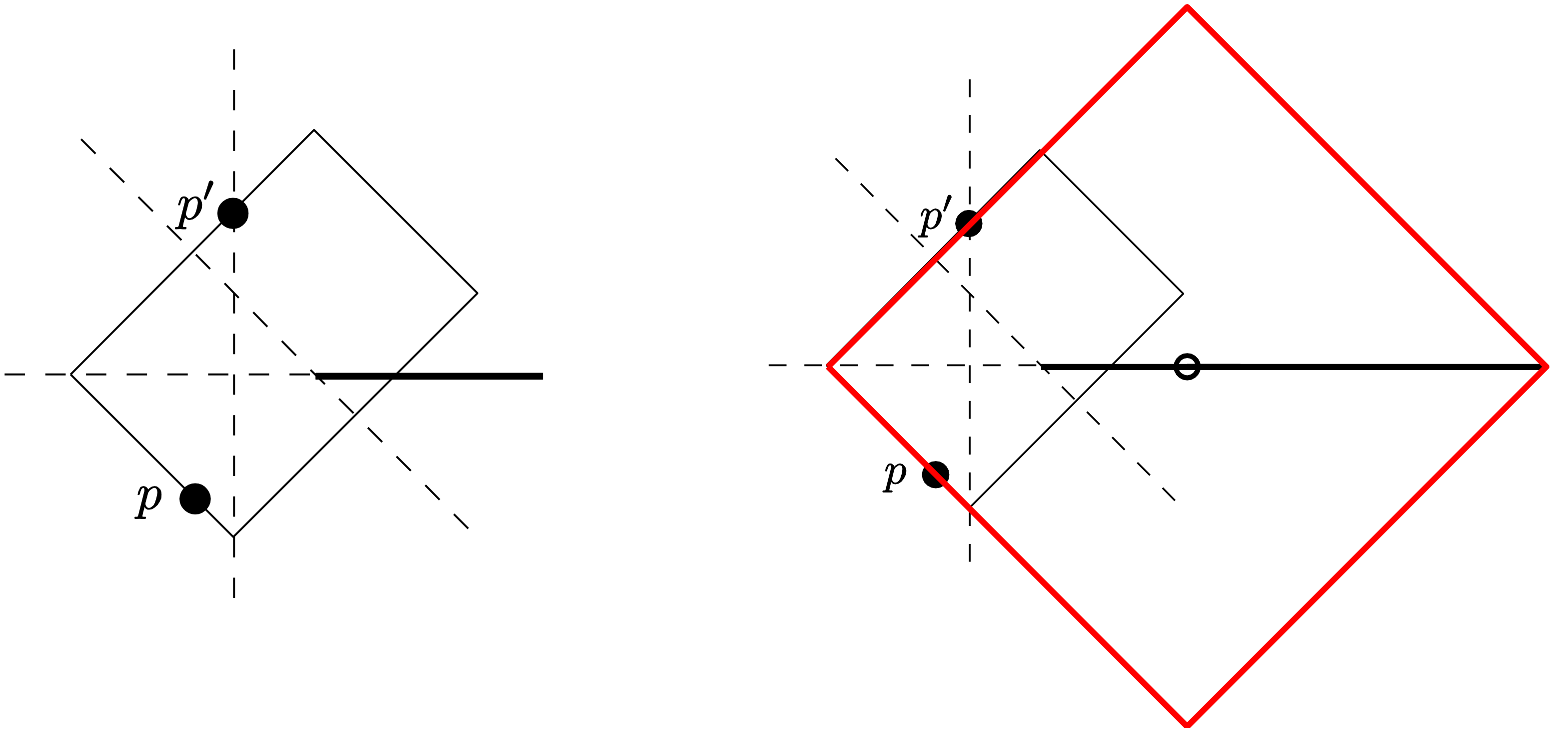
p′ dominates p



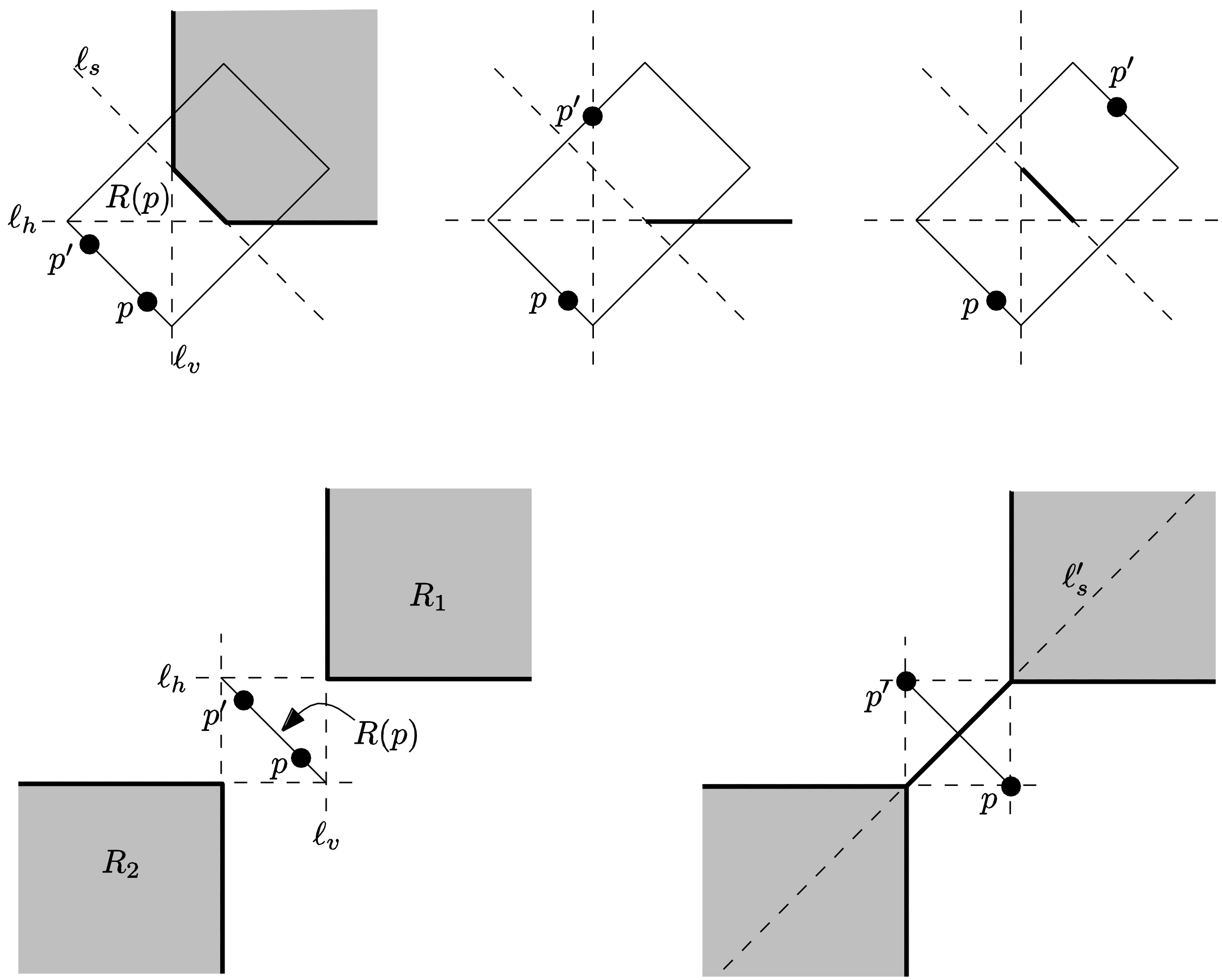
p′ dose not dominates p

Case (c)

The case that Q_1=Q



Case (c)



Algorithm

- **Input: P,Q**
- **Output: S**

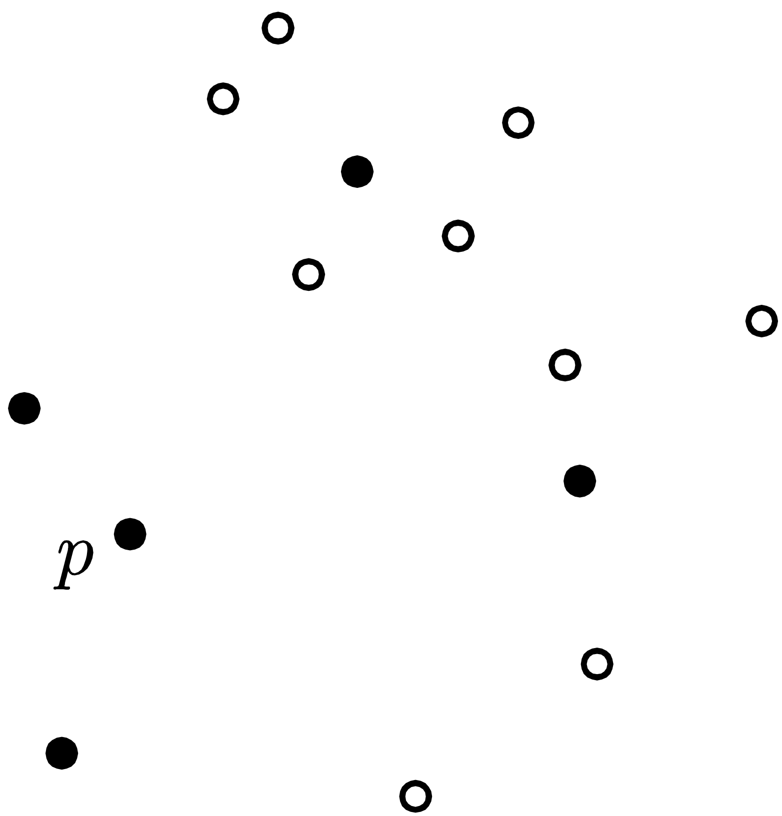
1. Compute

   a range counting query structure of P

   a segment dragging query structure of Q

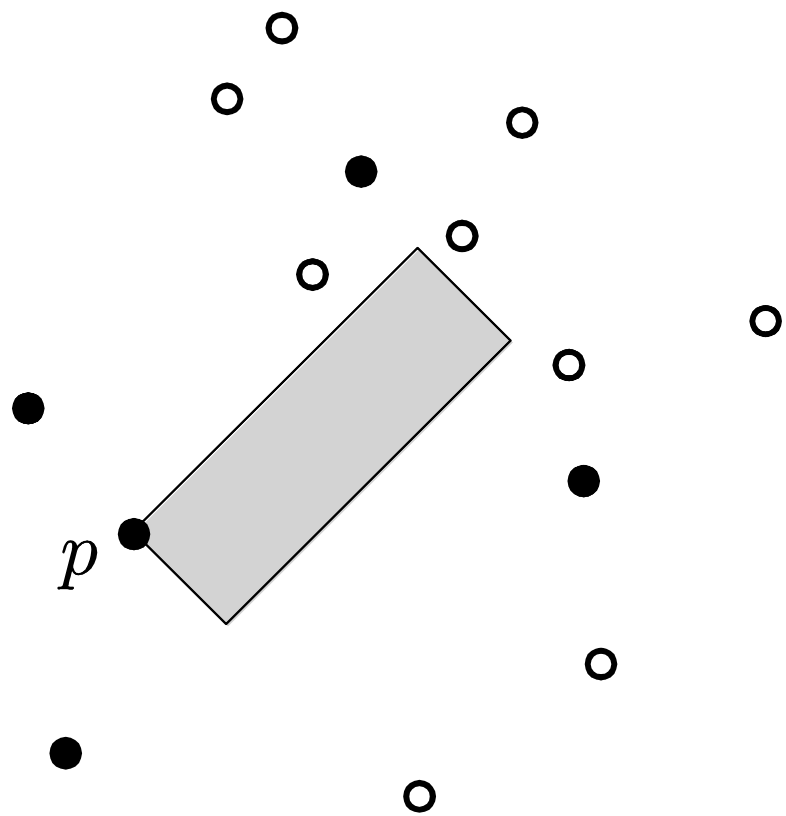
Algorithm

2. For each point p, compute R(p)



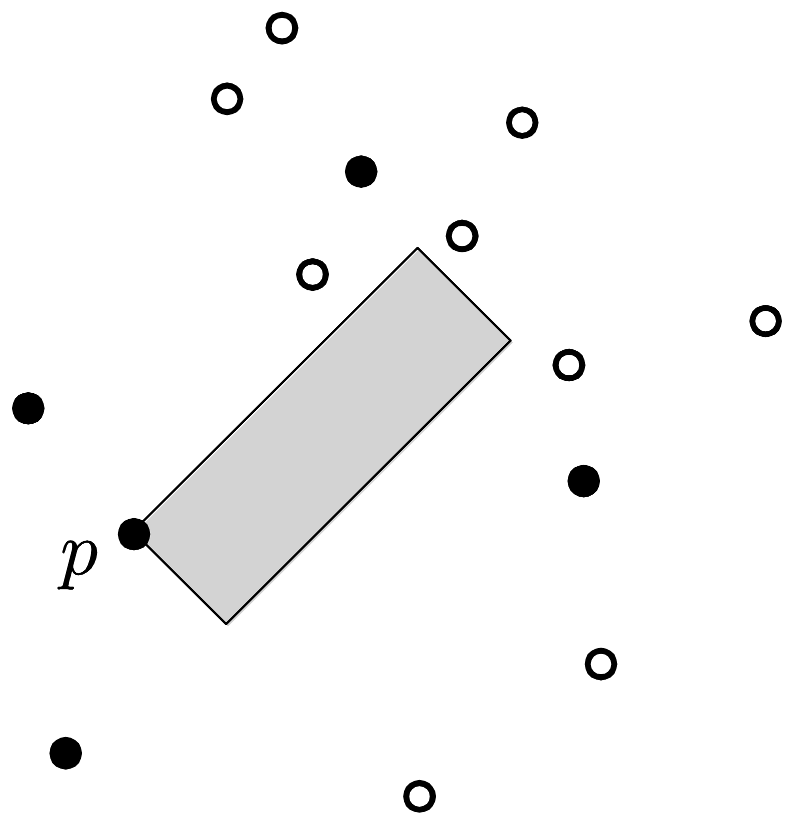
Algorithm

2. For each point p, compute R(p)



Algorithm

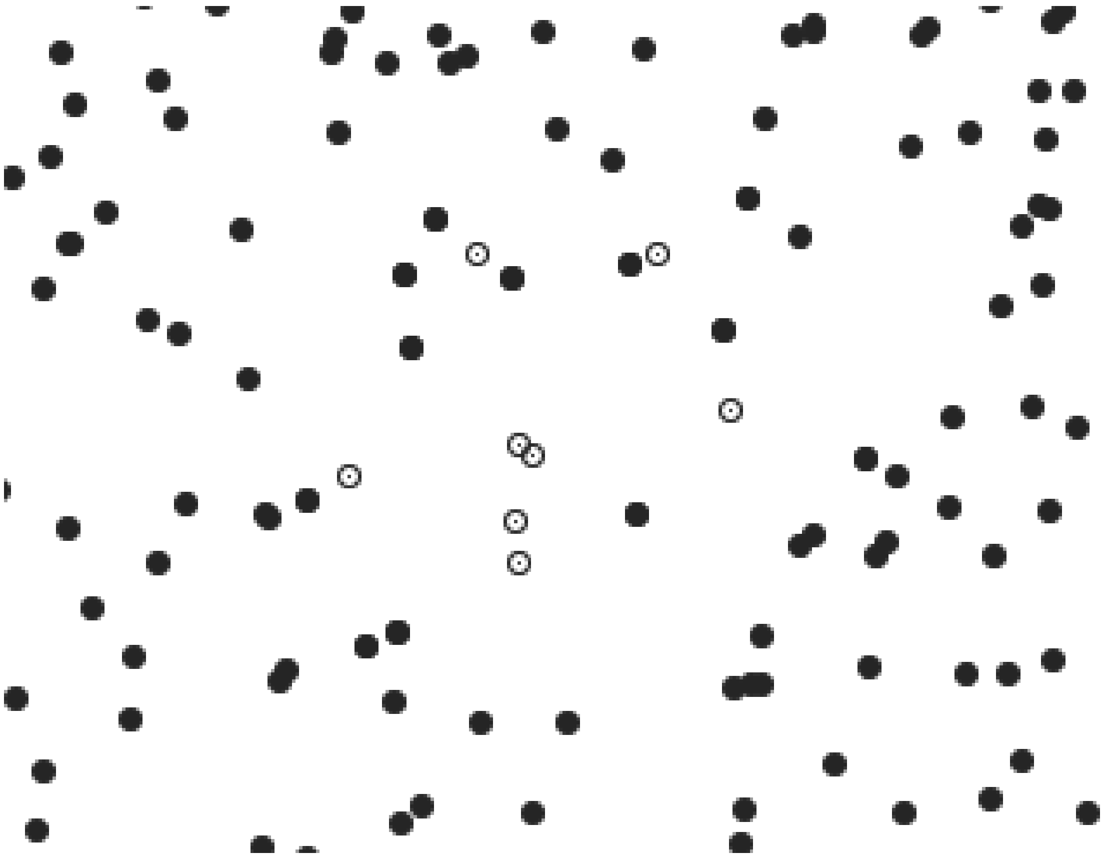
3. test whether p is skyline or not by the Range counting query for R(p)



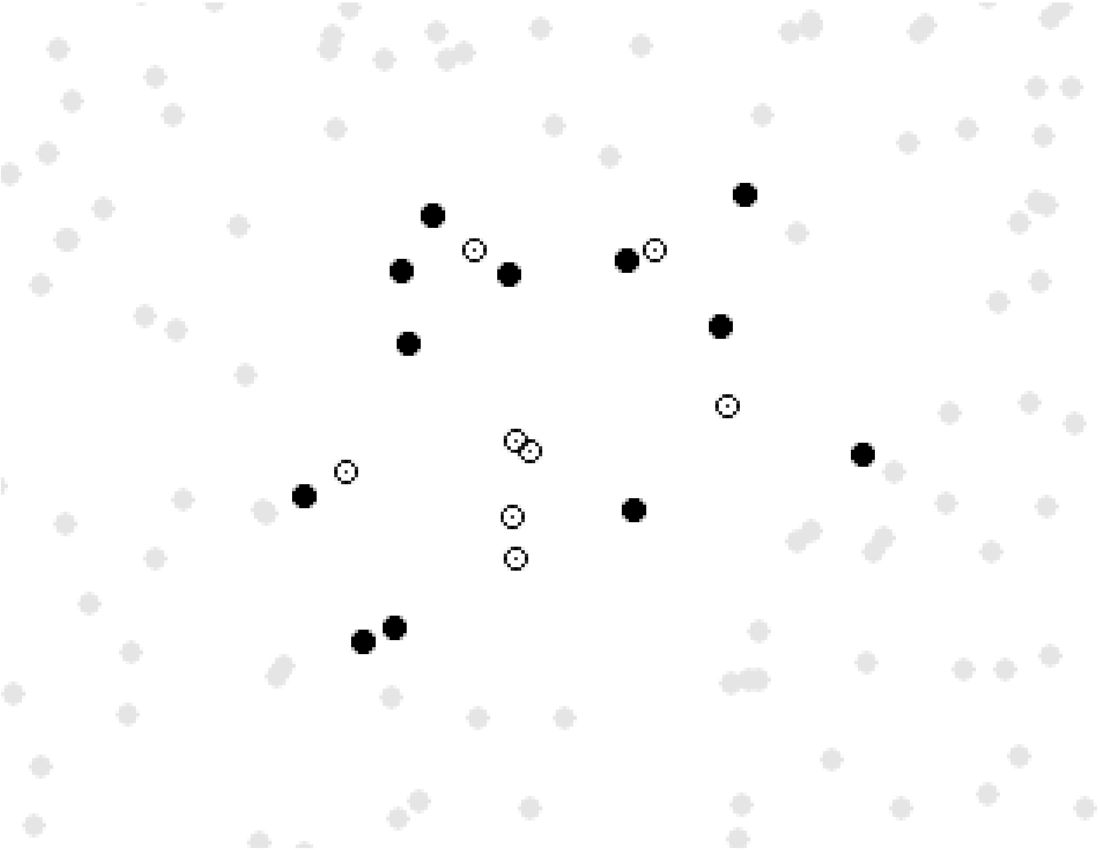
Analysis

- Complexity
  - Time : O(|P|log|P|)
  - Space : O(|P|)
- Avoid
  - pairwise comparison
  - operations for |Q| dimensinal data
- Easy to parallelize

Example



Example

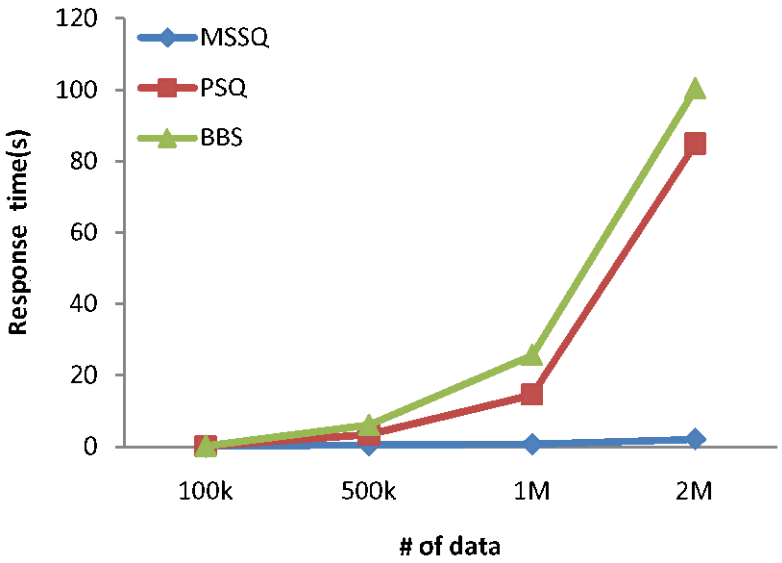


Experimental Evaluation

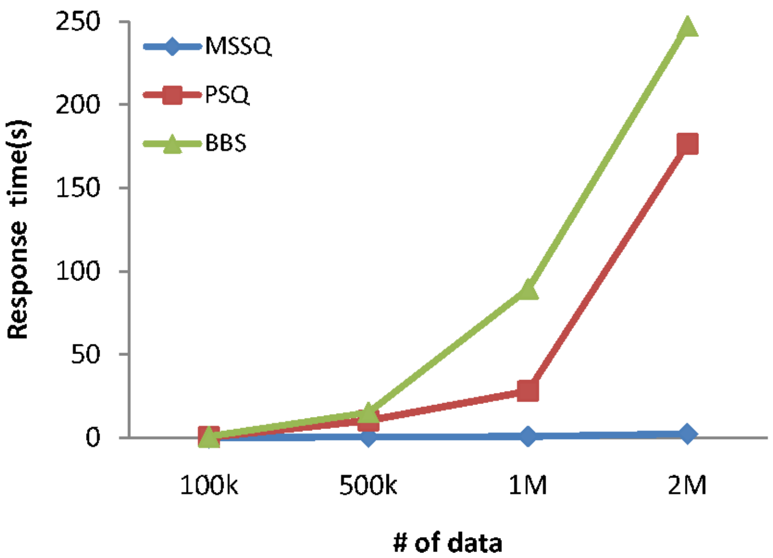
- Two base lines
  - PSQ : algorithm for skyline problems in general metric space.
  - BBS : algorithm for general skyline problems

Experimental Evaluation

- Synthetic datasets



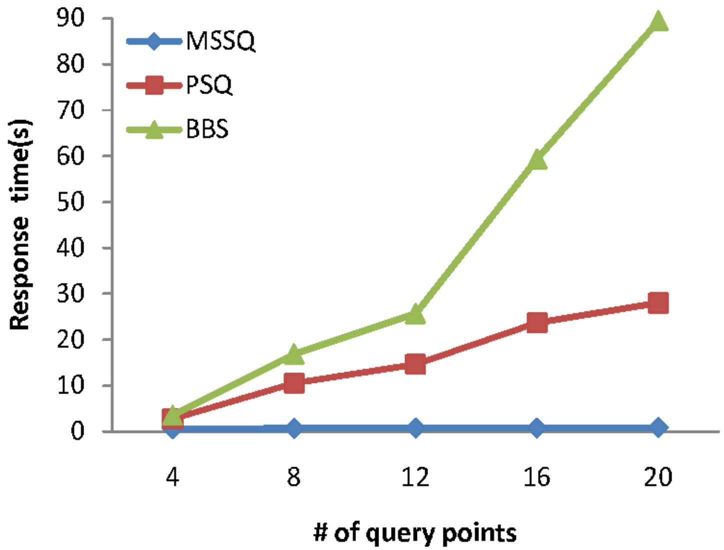
|Q|=12



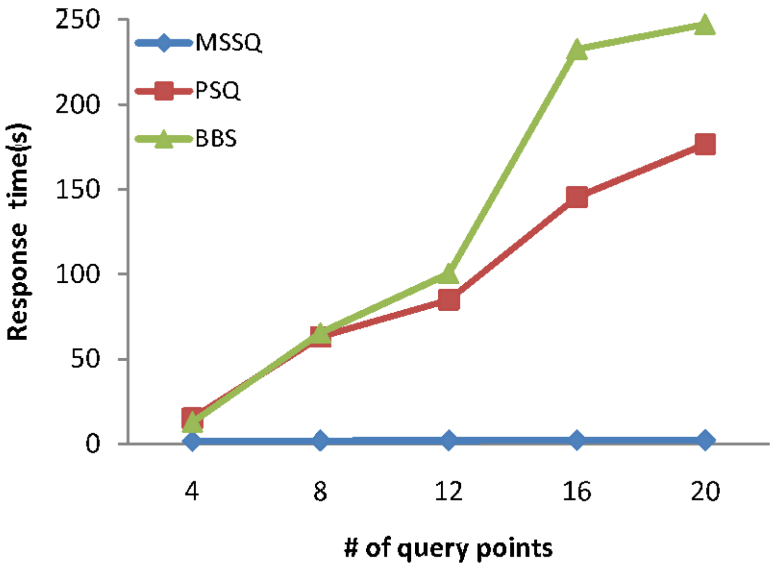
|Q|=20

Experimental Evaluation

- Synthetic datasets



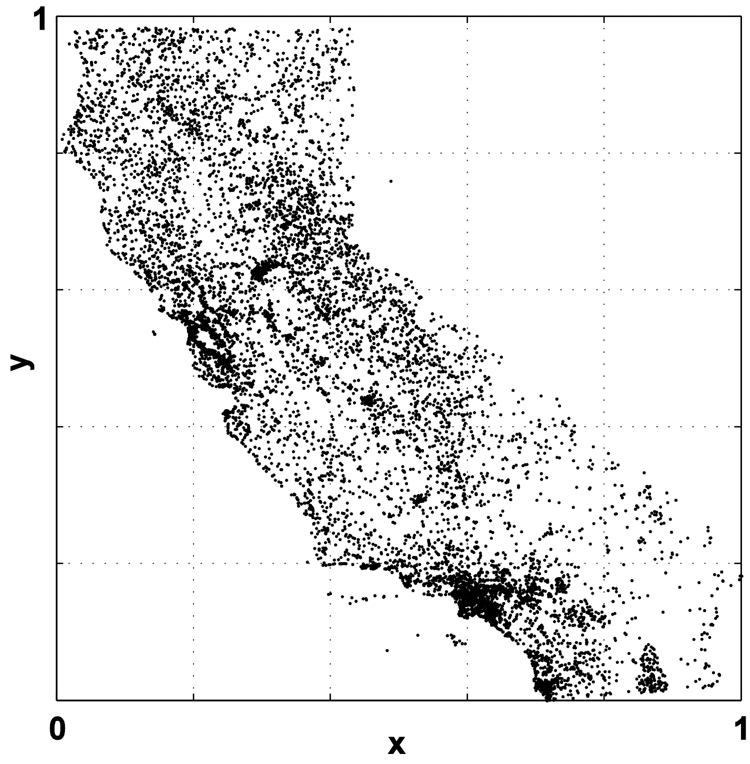
|P|=1M

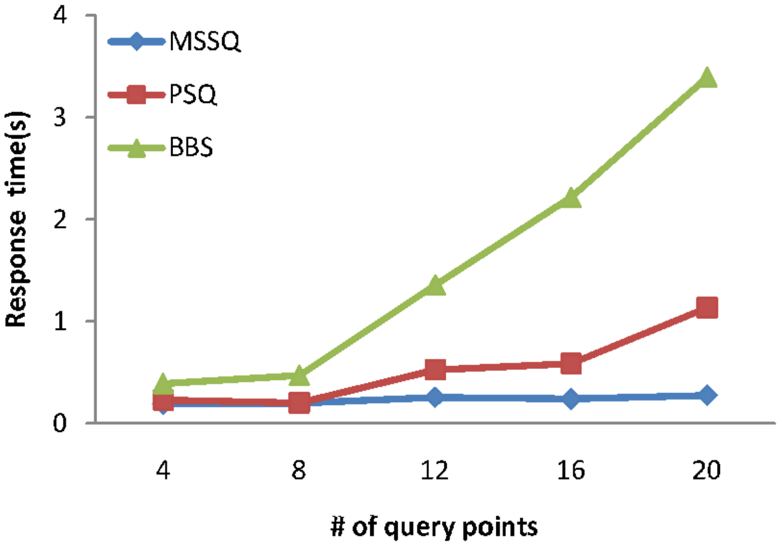


|P|=2M

Experimental Evaluation

- POI datasets : California POI data 10k





Q&A