On Dominating Your Neighborhood Profitably

Cuiping Li	Renmin University of China
Anthony K.H. Tung	National University of Singapore
Wen Jin	Simon Fraser University
Martin Ester	Simon Fraser University

Outline

- Motivation
- Problem Statements
- Symmetrical Methods
- Asymmetrical Methods
- Experimental Results
- Conclusion

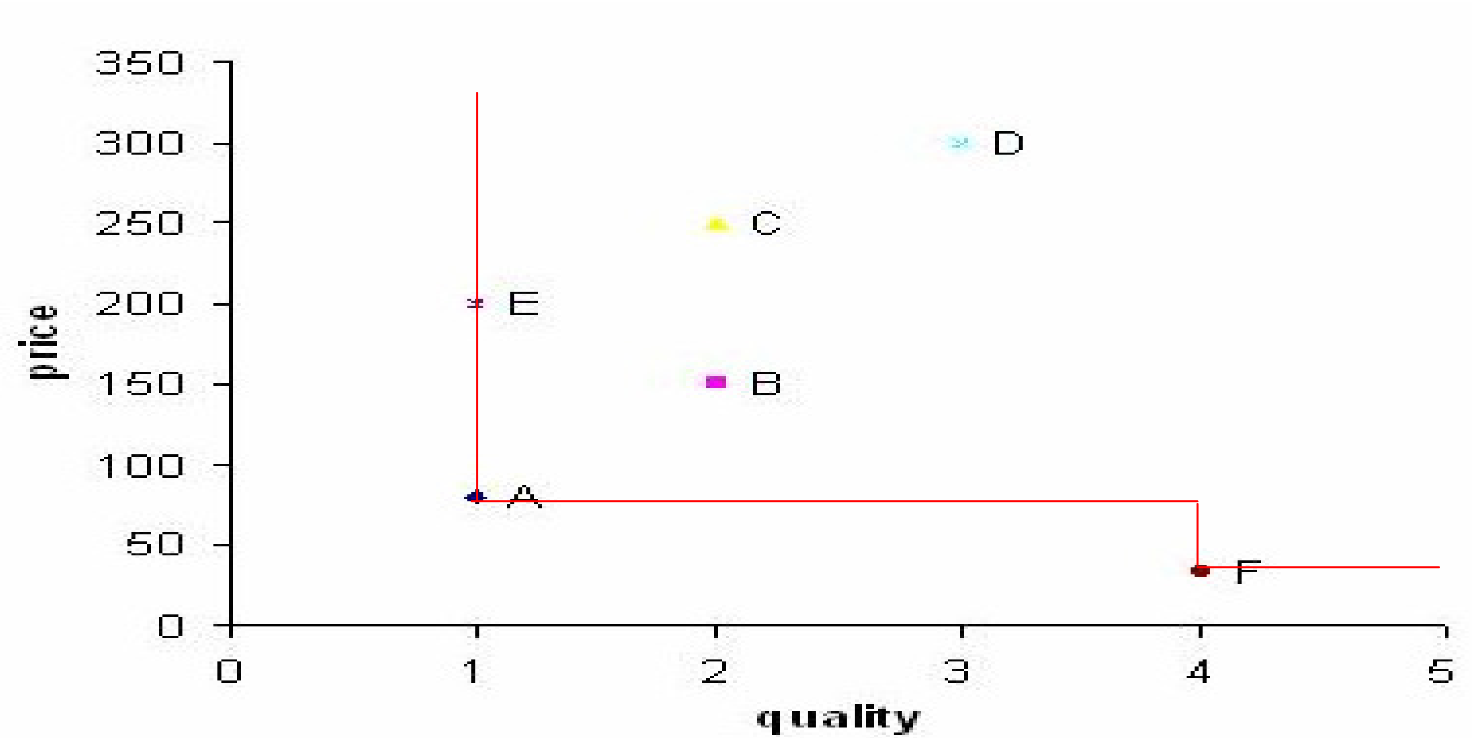
Definition of Dominate

- [Koss02] A point p dominates another point q, if
  - p is not worse than q in all dimensions
  - p is better than q in at least one dimension

- Assumption in this talk:
  - p is better than q in a dimension if p's value is less than q for that dimension

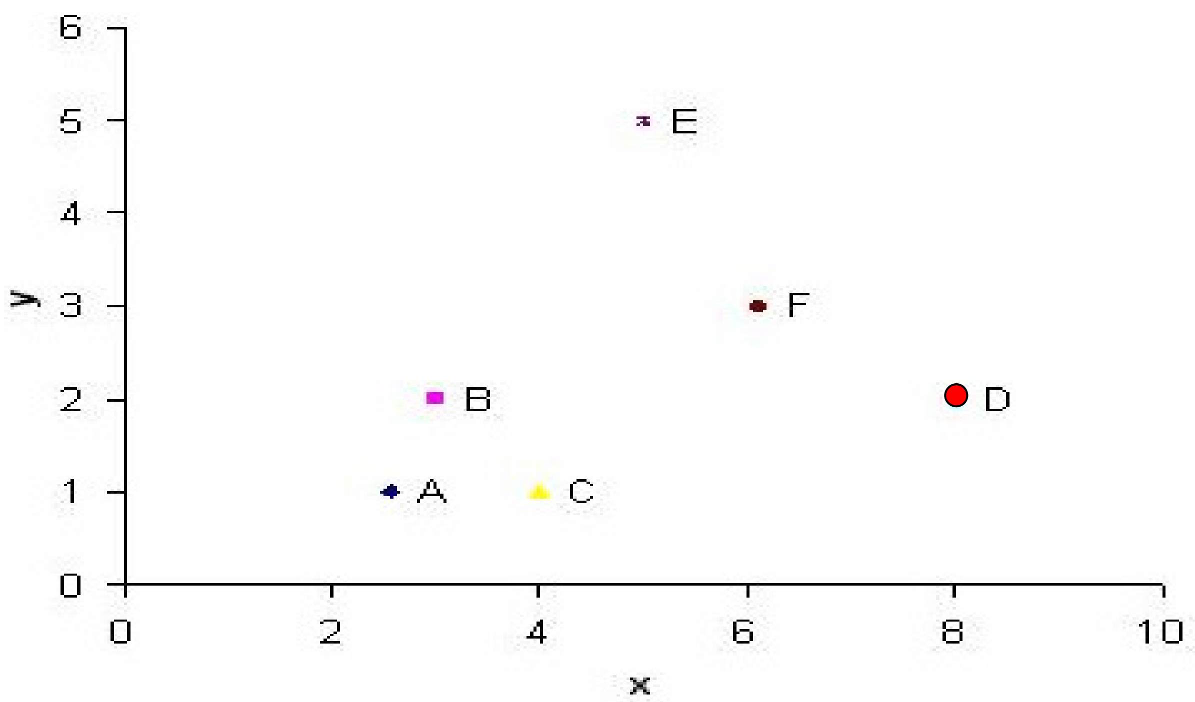
Definition of Skyline

- Example: Hotel (price, Quality)



- The skyline of a data set contains all the points not dominated by any other point

spatial location



Two Kinds of attributes

- Unlike the quality and price, the attribute x or y can not be said to be good or better if its value is small or large.

- To distinguish these two types of attributes
  - min/max attributes: such as quality and price
  - Spatial attributes: such as x and y

Perspective of Management

- The objective of a hotel manager:
  - to maximize the price (and consequently, the profit) for a given quality within certain constraints
    - Price and quality of competing hotels
    - The distance to the competing hotels

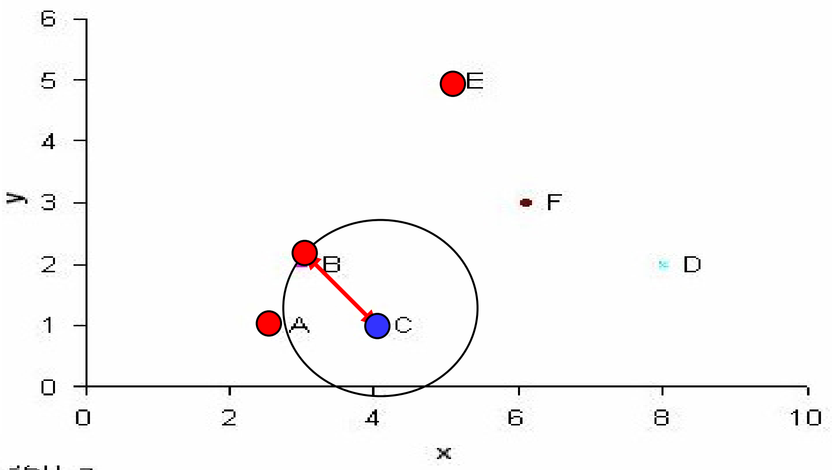
Outline

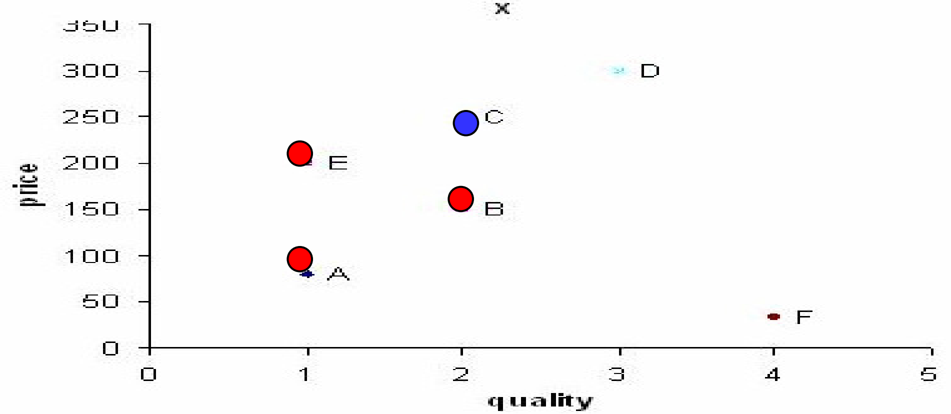
- Motivation
- Problem Statements
- Symmetrical Methods
- Asymmetrical Methods
- Experimental Results
- Conclusion

NDQ

- Nearest Dominators Query
  - Motivation
    - Hotel manager may want to ask: For my hotel q at location (x, y), what is the nearest hotel p that dominates q in the min/max dimensions?

NDQ



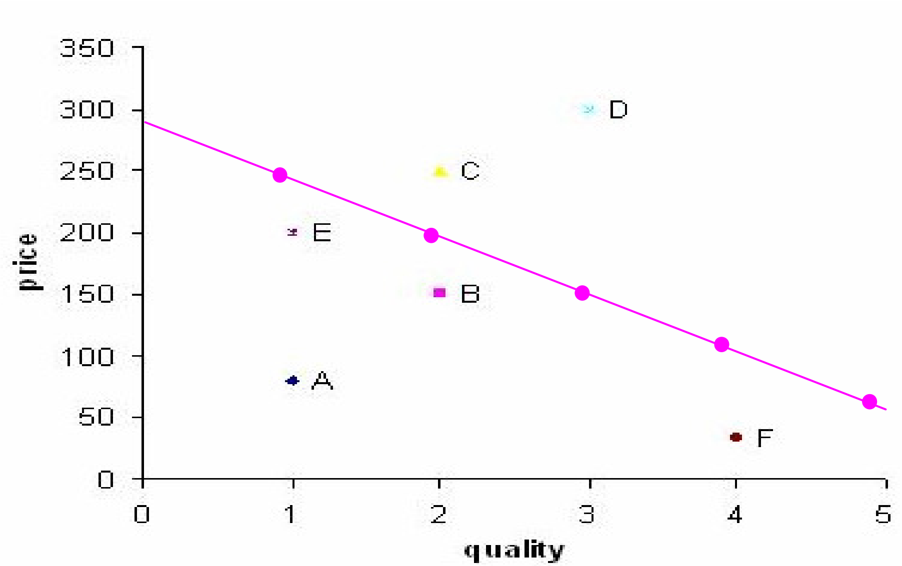


- ND(C) = B
- ndd(C)

Given any arbitrary object q in H, find its nearest dominator ND(q)

LDPQ

- Least Dominated, Profitable Points Query
  - Motivation
    - Hotel manager may want to ask: which hotel *q* is profitable while having the largest distance to its nearest dominator?



- *Since ndd(D) > ndd(C), hotel D is the answer*

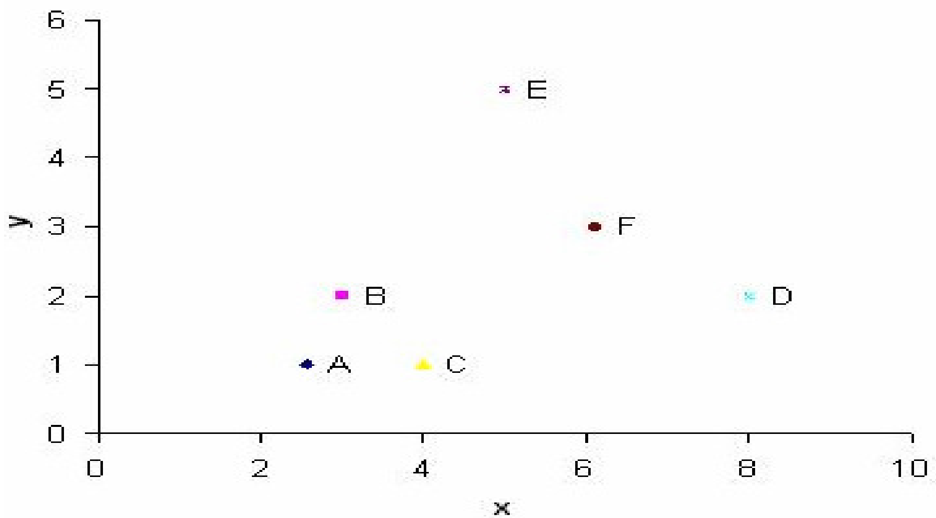
LDPQ

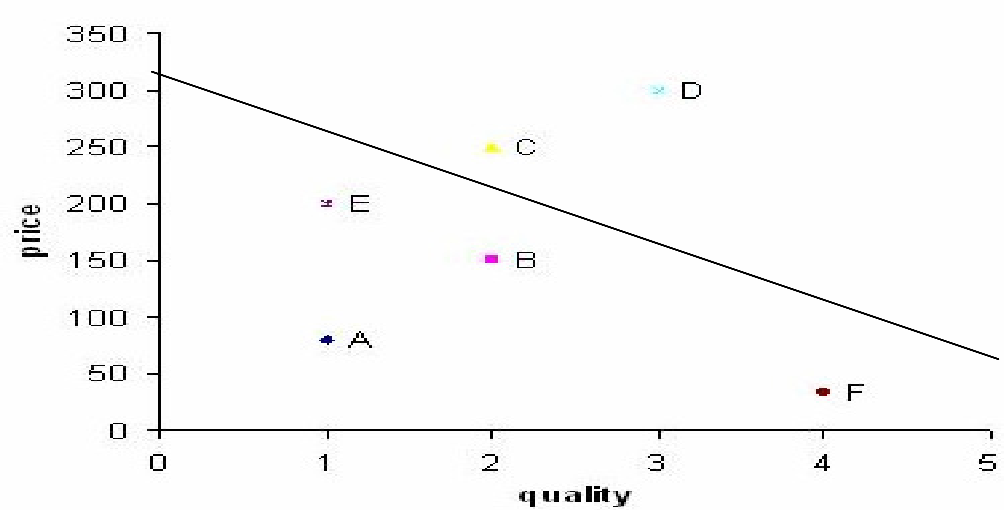
- Definition:
  - Given a dataset H and a hyper plane P, find the point t, which satisfies:
    - t is profitable
    - ndd(t) is the largest among all profitable points

ML2DQ

- Minimal Loss and Least Dominated Points Query
- Definition:
  - Given a profitability constraint and a distance threshold δ, find a hotel q such that:
    - $ndd(q) \geq \delta$
    - the difference between the price charged and the minimal profitable price is the smallest

Example for ML2DQ





- ndd(A) = ∞
- ndd(B) = 1.1
- ndd(E) = 4.6

- Assume δ=4.5

E will be returned

Neighborhood Dominant Queries

- NDQ \ LDPQ \ ML2DQ
  - A Family of query types considering the relationship between min/max and spatial attributes.

- two alternative query processing methods
  - Symmetrical
  - Asymmetrical

Outline

- Motivation
- Problem Statements
- Symmetrical Methods
- Asymmetrical Methods
- Experimental Results
- Conclusion

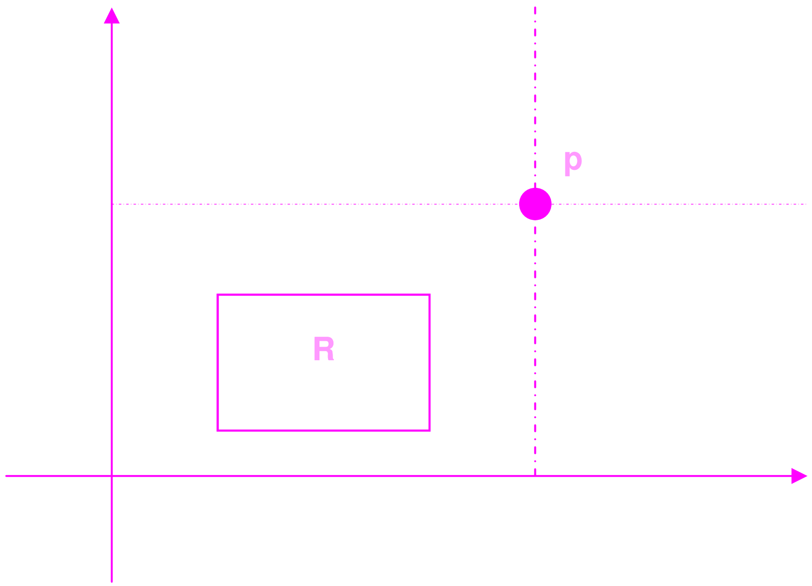
Symmetrical Methods

- treat min/max, spatial attributes as equal
- index them together in one R-Tree

Dominant Relationship (for NDQ)

- The dominant relationships between an MBR R and a given point p can be classified into three cases:
  - if $R_{ui} \le p_i$ for all min/max attribute I, then

    all points from R definitely dominate p

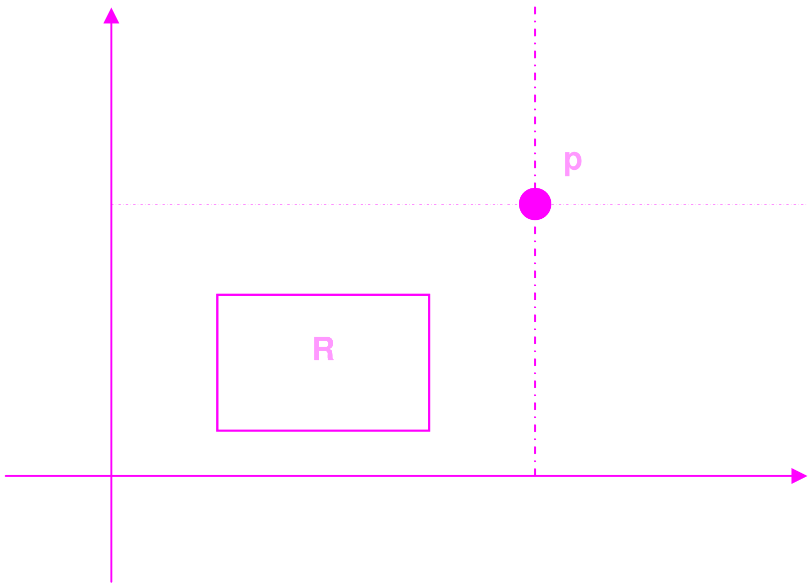


Dominant Relationship (for NDQ)

- The dominant relationships between an MBR R and a given point p can be classified into three cases:
  - if $R_{li} \le p_i$ for all min/max attribute i,

    $R_{uj} < p_j$ for |D|-1 min/max attributes j

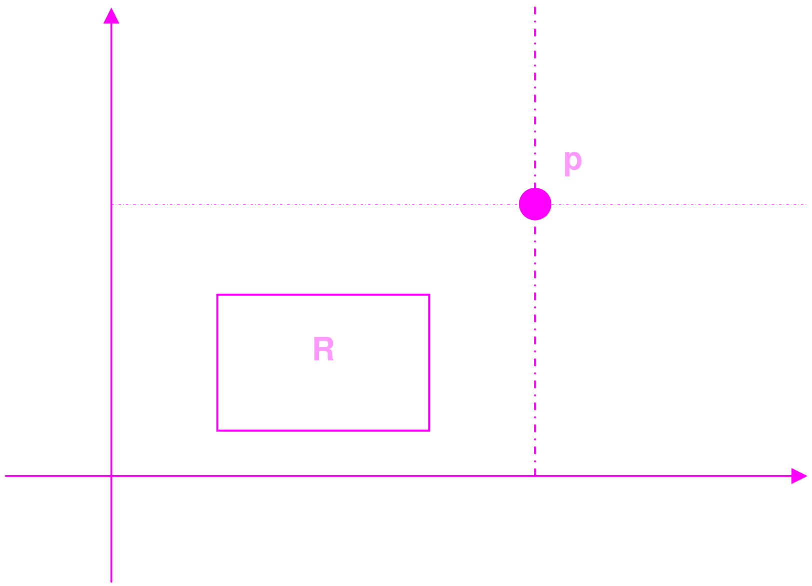
    then some points from R definitely dominate p



Dominant Relationship (for NDQ)

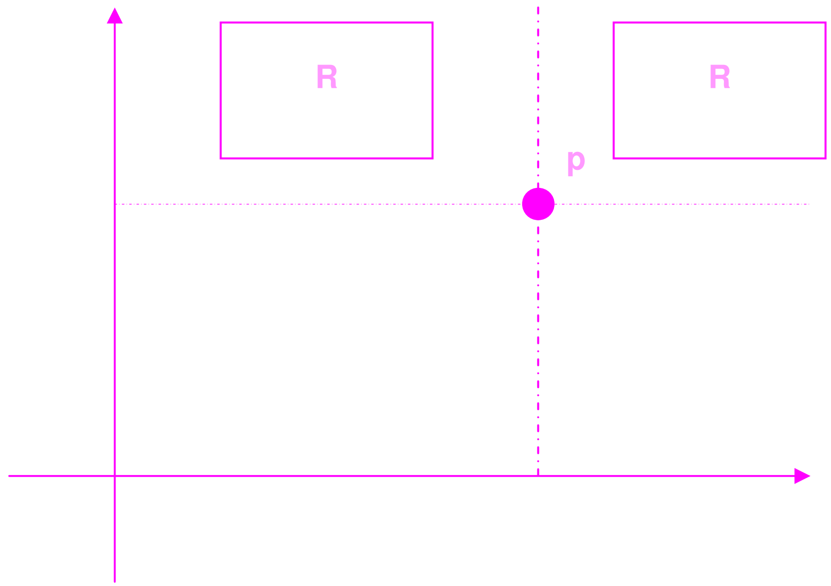
- The dominant relationships between an MBR R and a given point p can be classified into three cases:
  - if $R_{li} \leq p_i \leq R_{ui}$ for all min/max attribute I,

    then some points from R could dominate p



Dominant Relationship (for NDQ)

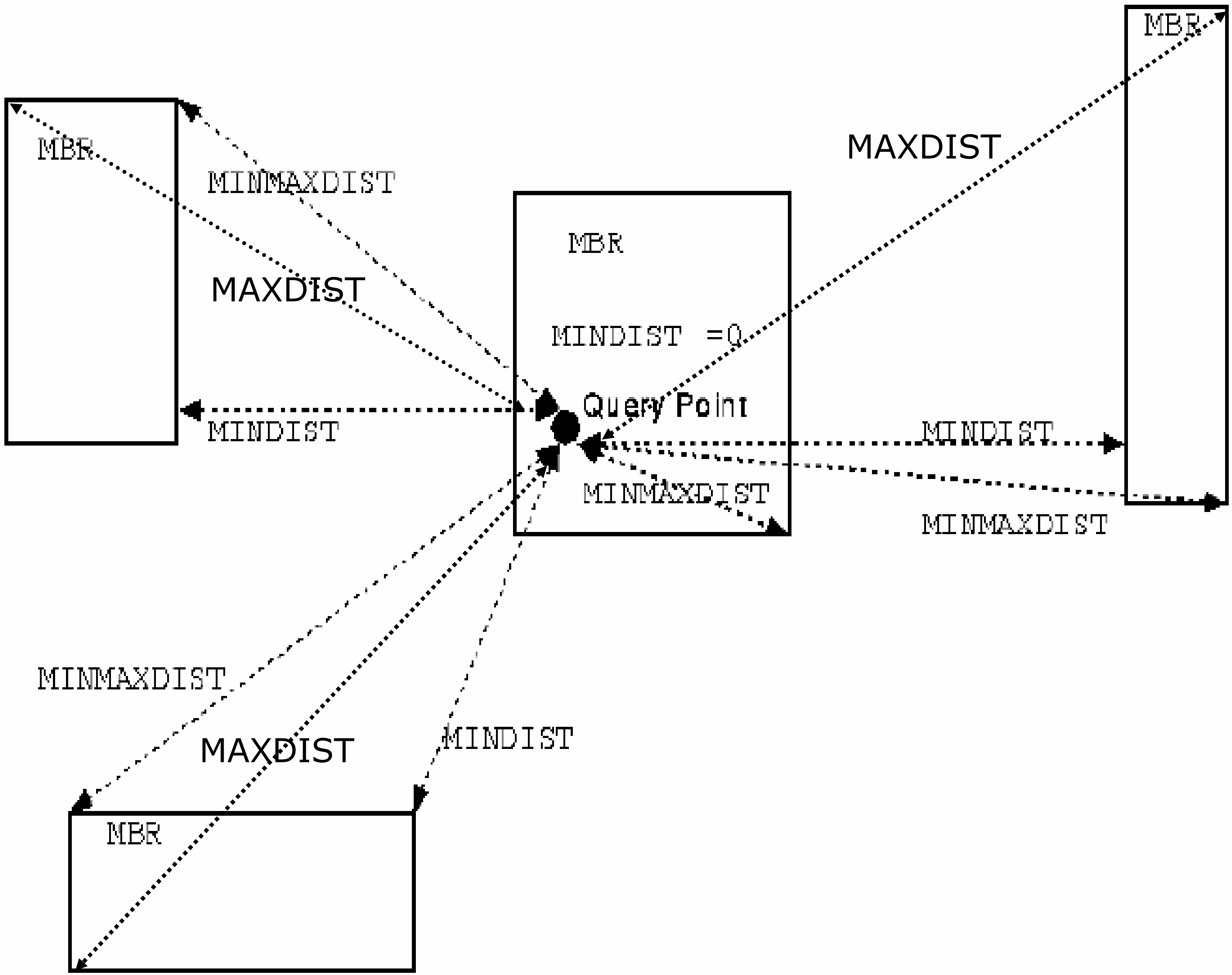
- The dominant relationships between an MBR R and a given point p can be classified into three cases:
  - Other cases: there does not exist dominant relationship between R and p



Spatial Relationship (for NDQ)

- Use three metrics to describe the distance between a MBR R and a point p
  - MINDIST(p,R): the nearest distance between p and any point in R
  - MAXDIST(p,R): the furthest distance between p and any point in R
  - MINMAXDIST(p,R): minimized distance upper bound that guarantee R contains at least one point which can dominate p.

Note: These metrics are computed using only spatial attributes

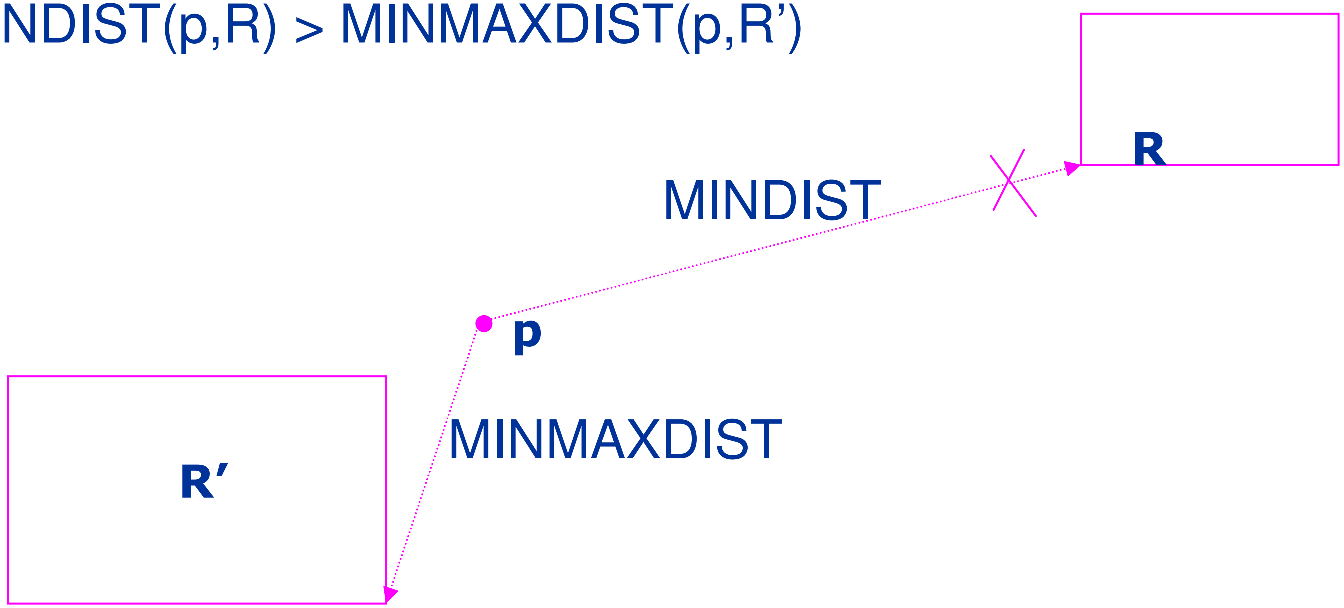

MBR
MINMAXDIST
MAXDIST
MINDIST
MBR
MINDIST =0
Query Point
MAXDIST
MBR
MINDIST
MINMAXDIST
MINMAXDIST
MINMAXDIST
MAXDIST
MINDIST
MBR

Best First Traversal Algorithm

- Start from the root MBR of R-tree, place its children MBRs into the heap
- Within the heap, order MBRs by:
  - Case 3, case 2, case 1
  - MINDIST, ascending
- Beginning from the top MBR of the heap, recursively extracting children of MBRs, and inserting those potential dominators of p into the heap.
- Algorithm terminated when the heap empty

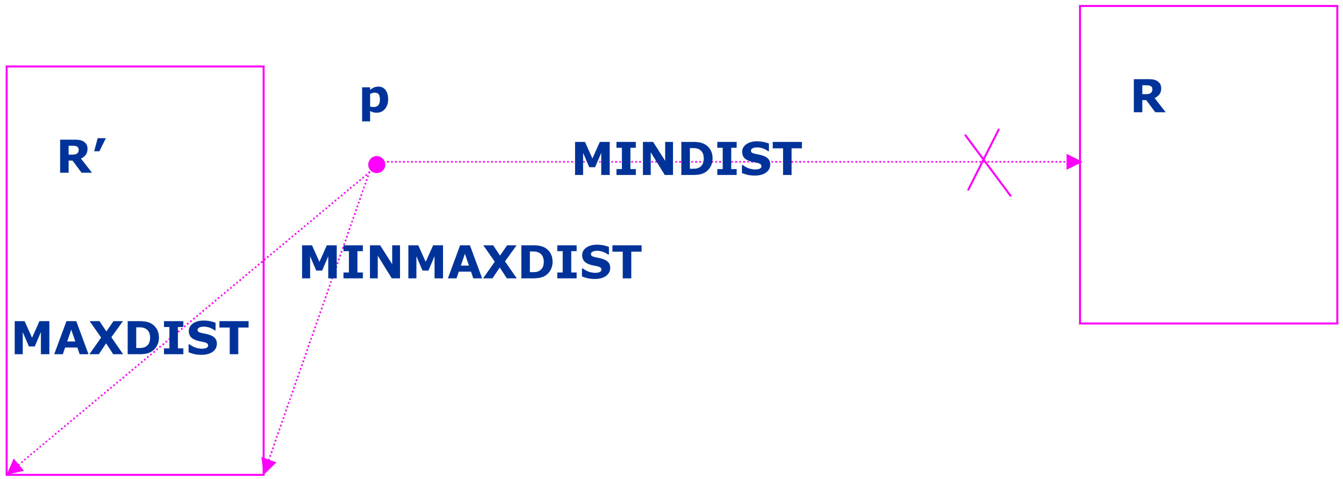
Pruning Strategy 1 (for NDQ)

- An MBR R is discarded if there exists an R' s.t.
  - p and R' correspond to case 3
  - MINDIST(p,R) > MINMAXDIST(p,R')



Pruning Strategy 2 (for NDQ)

- An MBR R is discarded if there exists an R' s.t.
  - p and R' correspond to case 2
  - MINDIST(p, R) > MAXDIST(p, R')



Why not use MINMAXDIST in case 2?

Can not ensure there exists a dominator in this distance

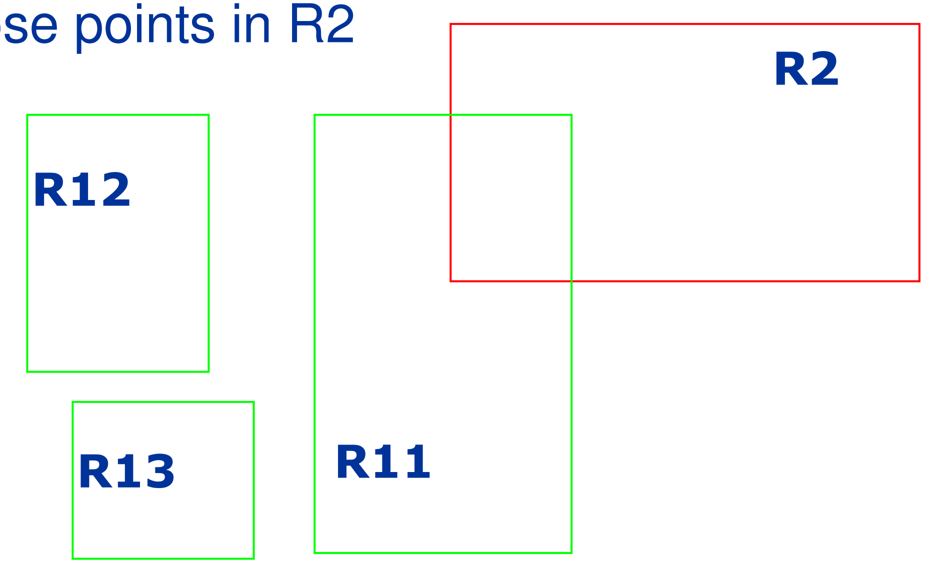
LDPQ with Symmetrical R-tree

- Naïve method:
  - First, perform a NDQ search for all points in the profitable region
  - Second, select the point with the largest nearest dominator distance

- More efficient method:
  - merge above two steps into one

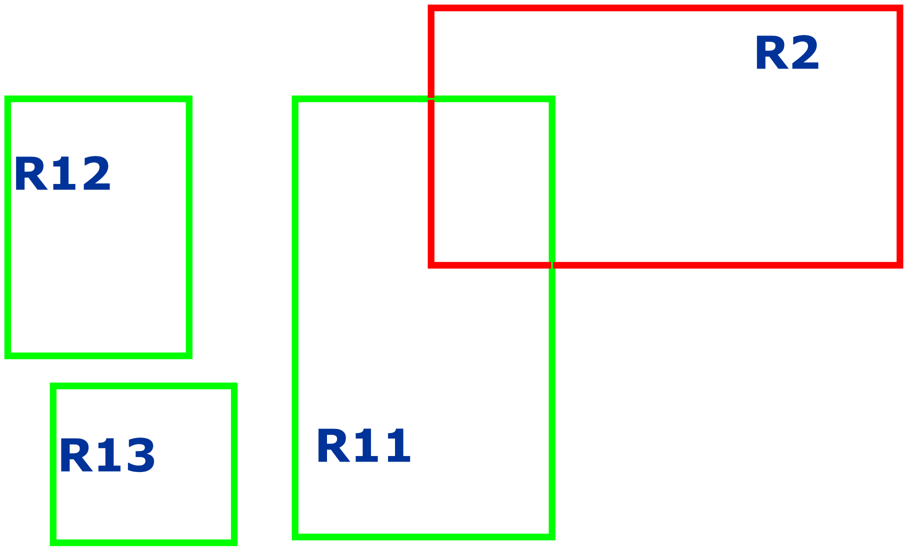
LDPQ with Symmetrical R-tree

- Monitor two types of MBRs
  - PdMBR: MBRs that are potentially dominated by some points and are candidates for the output answers
    - Any MBR in the R-tree can be PdMBR unless it is pruned
  - For each PdMBR R2,
    - PnrMBR: MBRs that potentially contain the nearest dominators for those points in R2



LDPQ with Symmetrical R-tree

- The dominant relationship between MBRs from PdMBR and PnrMBR can be following:
  - Case1 : some points from R1 could dominate some points from R2
  - Case 2: some points from R1 definitely dominate all points from R2
  - Case 3: all points from R1 definitely dominate all points from R2



Another three useful Metrics

- MINMINDIST(R1,R2)
- MAXMAXDIST(R1,R2)
- MAXMINMAXDIST(R1,R2)
    - … details can be referenced in the paper

Another three useful Metrics

- MINMINDIST(R1,R2)

$$\min_{p \in CORNER(R2)} MinDist(R1, p)$$

- MAXMAXDIST(R1,R2)

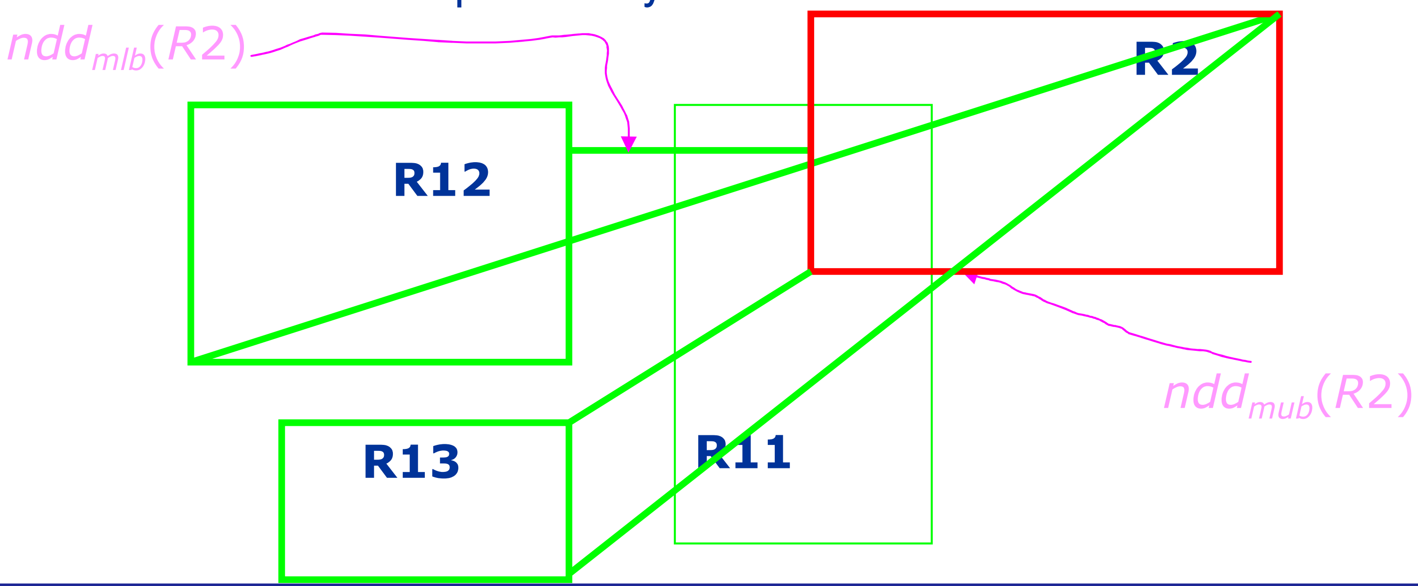
$$\max_{p \in CORNER(R2)} MaxDist(R1, p)$$

- MAXMINMAXDIST(R1,R2)

$$\max_{p \in CORNER(R2)} MinMaxDist(R1, p)$$

Two Thresholds for Pruning

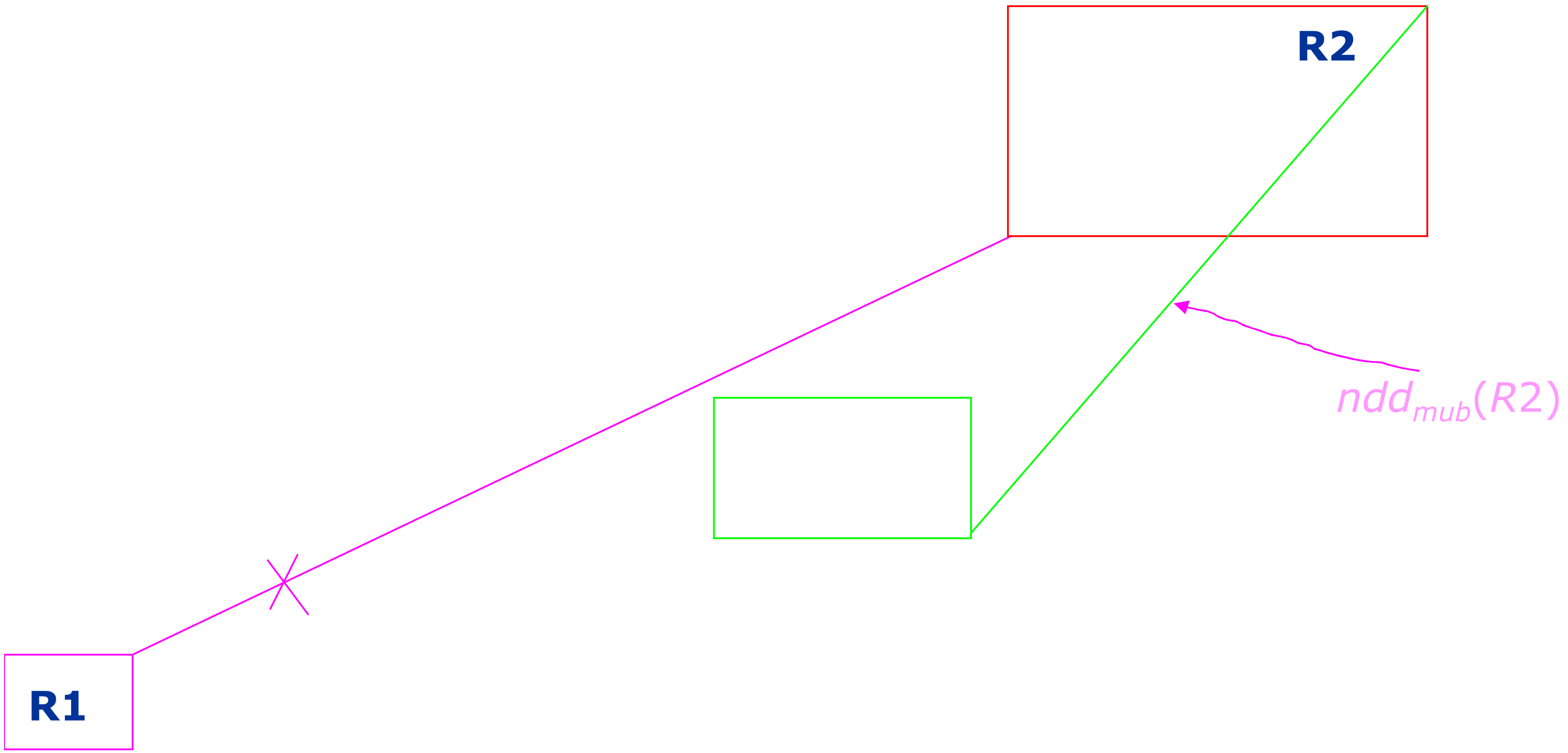
- For each PdMBR R2, maintain two variables:
  - $ndd_{mlb}(R2)$: minimum lower bound distance between R2 and its PnrMBRs
    - case 3 or case 2: updated by MINMINDIST
  - $ndd_{mub}(R2)$: minimum upper bound distance between R2 and its PnrMBR
    - guarantee there is at lease one point can dominate all points in R2
    - case3: updated by MAXMINMAXDIST
    - case2: updated by MAXMAXDIST



Local Pruning (for LDPQ)

- Given R2, R1 can be removed from PnrMBR(R2) if:

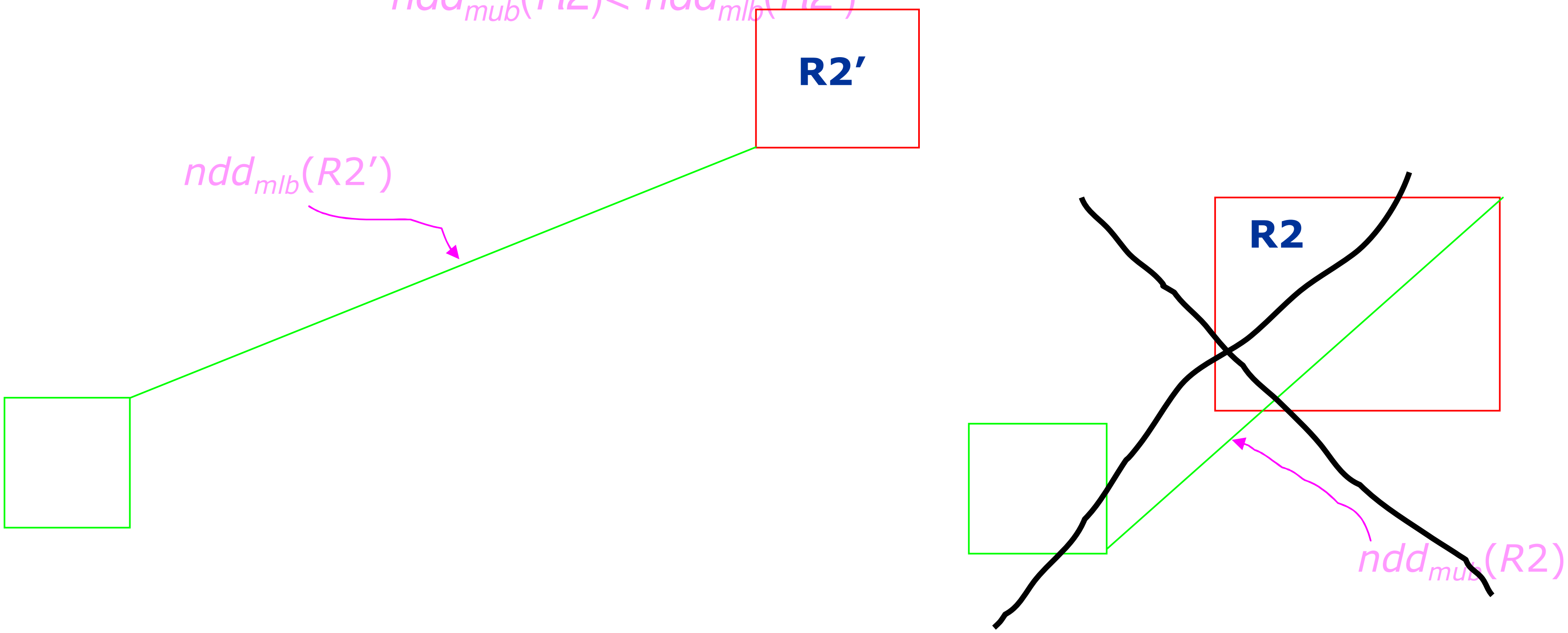
$$MINMINDIST(R1,R2) > ndd_{mub}(R2):$$



Global Pruning (for LDPQ)

- Any R2 can be removed from PdMBR if there exists a R2' s.t.

$ndd_{mub}(R2) < ndd_{mlb}(R2')$



ML2DQ with Symmetrical R-tree

- The aim of this type query is:
  - to find a point q in the unprofitable region s.t.:
    - the distance to P is minimized
    - $ndd(q) \geq \delta$
- To process this type query:
  - Adopt the same best first search approach as LDPQ
  - Pruning strategies:
    - Only considering the MBRs intersecting the non-profitable region
    - R1 is removed if $ndd(R1) < \delta$
    - R1 is removed if R1 is far away from P

Outline

- Motivation
- Problem Statements
- Symmetrical Methods
- Asymmetrical Methods
- Experimental Results
- Conclusion

Asymmetrical Methods

- Spatial attributes and min/max attributes play different roles when query is processed.

- The whole process includes two steps:
  - Clustering into micro-cluster (spatial attributes)
  - Constructing a Asymmetrical R-Tree (min/max attributes), and associate the spatial info with the min/max info

The First Step

- Clustering into micro-cluster
  - Points are clustered into k micro-clusters by spatial attributes
  - Finished by a typical pre-processing algorithm BIRCH
  - Each micro-cluster MCi, has:
    - Cluster id: i
    - Mean value: MCi.m
    - Radius: MCi.r

The Second Step

- Constructing an Asymmetrical R-Tree
  - MBRs are formed by min/max attributes

- In order to capture the spatial info
  - Each MBR is associated with a bitmap of size k. each bit represents one micro-cluster
    - If some point of MCi appears also in the MBR, set bit i to 1, otherwise 0

NDQ with Asymmetrical R-Tree

- Given a query point p, and a micro-cluster MCi:
  - *MinDist(p,MCi) = max{dist(p,MCi,m) –MCi.r, 0}*
  - *MaxDist(p,MCi) = dist(p,MCi,m) +MCi.r*

- Based on this, redefine:
  - *MINDIST(R, p)*
  - *MAXDIST(R,p)*
  - *MINMAXDIST(R,p)*
    - *…details can be referenced in the paper*

NDQ with Asymmetrical R-Tree

- Given a query point p, and a micro-cluster MC_i:
  - *$MinDist(p,MC_i) = max\{dist(p,MC_i,m) - MC_i.r, 0\}$*
  - *$MaxDist(p,MC_i) = dist(p,MC_i,m) + MC_i.r$*

- Based on this, redefine:
  - *$MINDIST(R, p)=min\{MinDist(p, MC_{Ri}), MC_{Ri} \in MCin(R)\}$*
  - *$MAXDIST(R,p)=max\{MaxDist(p,MC_{Ri}), MC_{Ri} \in MCin(R)\}$*
  - *$MINMAXDIST(R,p)=min\{MaxDist(p,MC_{Ri}),MC_{Ri} \in MCin(R)\}$*
    - *Here, MCin(R) denote the set of micro-clusters that are mark as present in R*

LDPQ(ML2DQ) with Asymmetrical R-Tree

- Given any two micro-clusters MCi and MCj:
  - *MinDist(MCi,MCj) =max{dist(MCi.m, MCj.m)-MCi.r-MCj.r , 0}*
  - *MaxDist(MCi,MCj) =dist(MCi.m, MCj.m)+MCi.r+MCj.r*

- Based on this, redefine:
  - *MINMINDIST(R1,R2)*
  - *MAXMAXDIST(R1,R2)*
  - *MAXMINMAXDIST(R1,R2)*
    - *…details can be referenced in the paper*

LDPQ(ML2DQ) with Asymmetrical R-Tree

- Given any two micro-clusters MC_i and MC_j:
  - *MinDist*(MC_i, MC_j) $=max\{dist(MC_i.m, MC_j.m) - MC_i.r - MC_j.r\ ,\ 0\}$
  - *MaxDist*(MC_i, MC_j) $=dist(MC_i.m, MC_j.m) + MC_i.r + MC_j.r$

- Based on this, redefine:
  - $MINMINDIST(R1,R2)= min\{MinDist(MC_{R1i}, MC_{R2j})\}$
  - $MAXMAXDIST(R1,R2)= max\{MaxDist(MC_{R1i}, MC_{R2j})\}$
  - $MAXMINMAXDIST(R1,R2)=max\{MaxDist(MC_{R2i},\ NNMAX(MC_{R2i}, MC_{in}(R1)),$
    - Here, $MC_{R1i} \in MC_{in}(R1)$, $MC_{R2i} \in MC_{in}(R2)\}$
    - $NNMAX(MC_{R2i}, MC_{in}(R1)))\}$ *denote the micro-cluster in* $MC_{in}(R1)$ *which has the smallest MaxDist to* MC_{R2i}

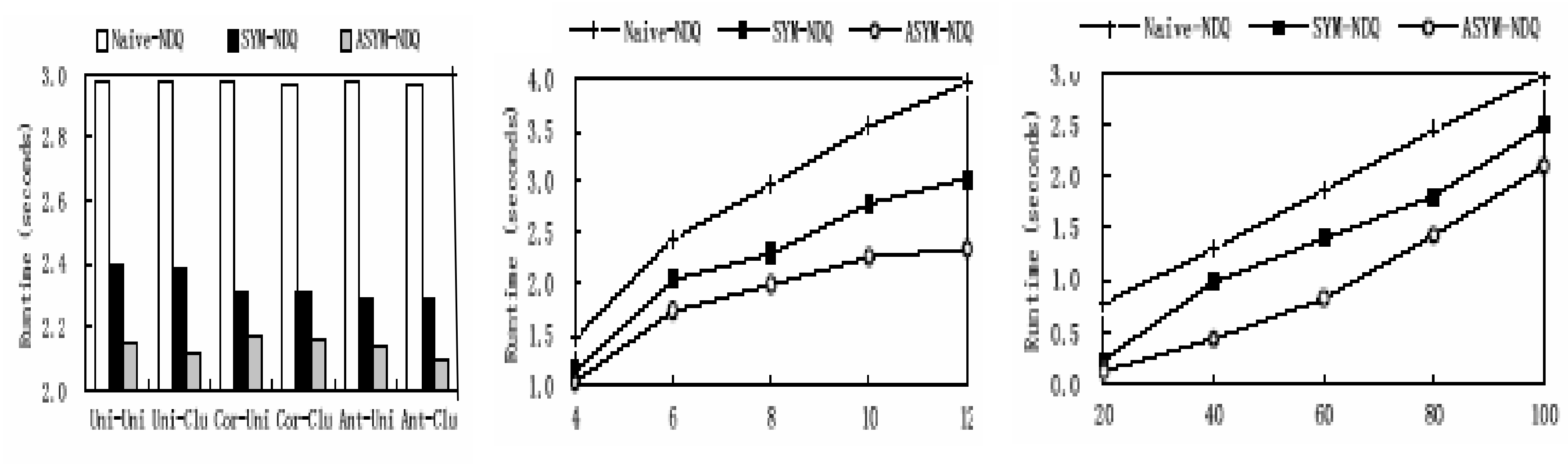
Outline

- Motivation
- Problem Statements
- Symmetrical Methods
- Asymmetrical Methods
- Experimental Results
- Conclusion

Experiment Results

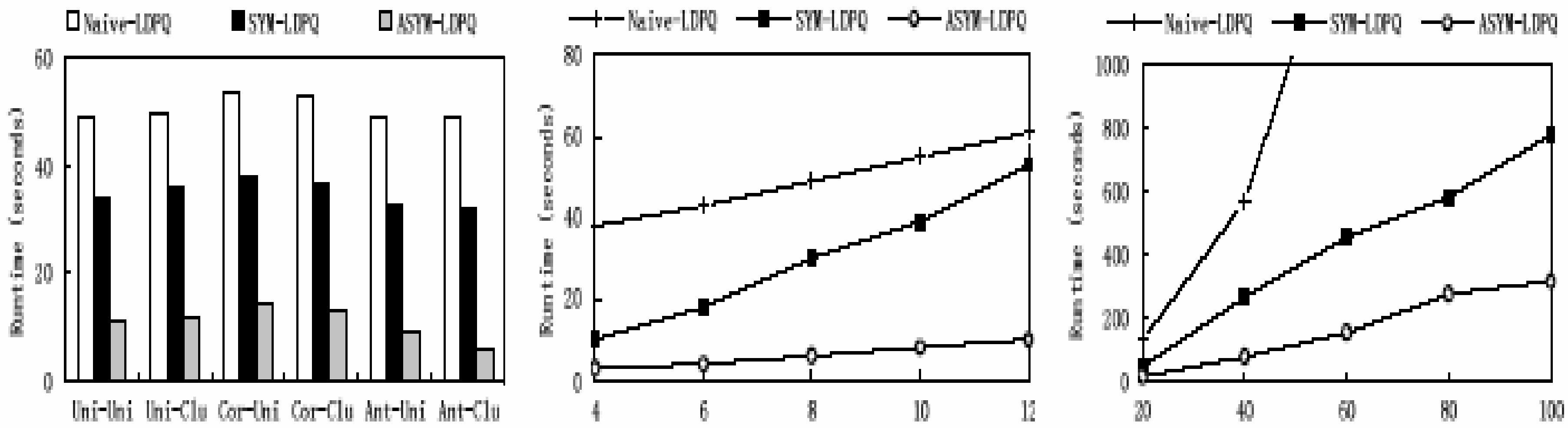
- Synthetic Data Set
  - Min/max attributes: Correlated, Independent, Anti-Correlated
  - Spatial attributes: uniform, clustered
- Query Type: NDQ, LDPQ, ML2DQ
- Query Process Algorithm: Naïve, Sym, ASym
- Default Values:
  - Dimensionality: 8
  - Data size: 100k
  - The number of micro-clusters: 50

Query Performance for NDQ



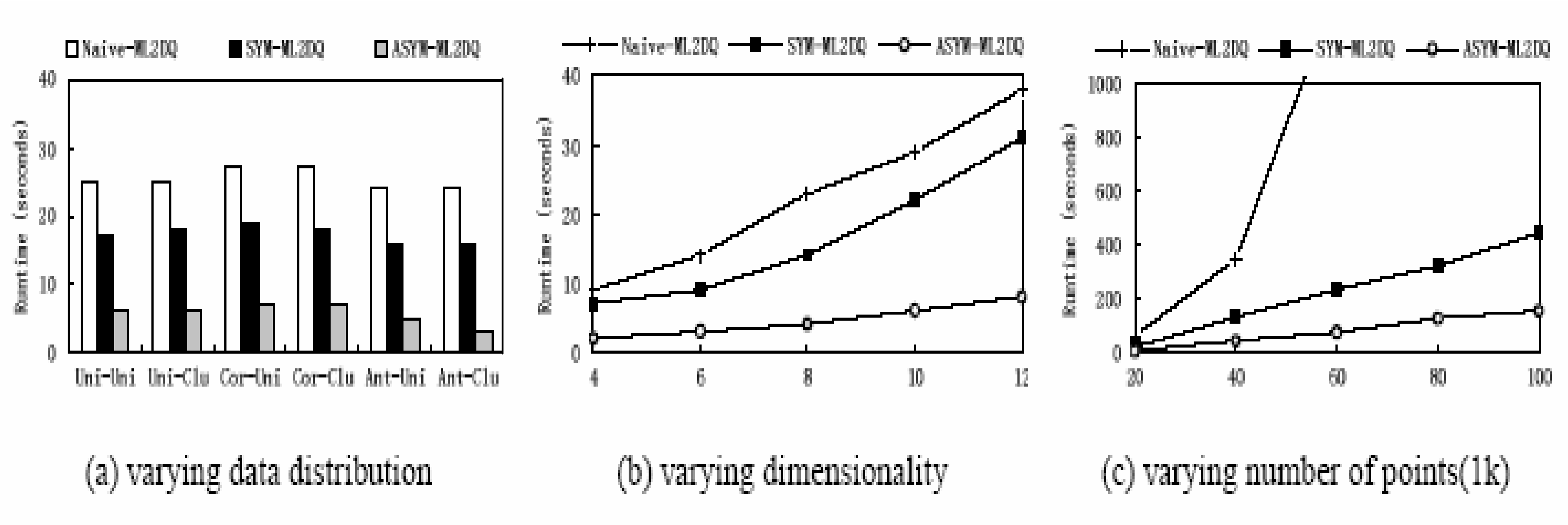
(a) varying data distribution (b) varying dimensionality (c) varying number of points(1k)

Query Performance for LDPQ



(a) varying data distribution (b) varying dimensionality (c) varying number of points(1k)

Query Performance for ML2PQ



Conclusion

- Present three novel types of skyline queries as representative for neighborhood dominant queries: NDQ\LDPQ\ML2DQ. Exploit not only min/max attributes but also spatial attributes

- Based on standard or extended index structures, propose symmetrical as well as asymmetrical methods to process the queries

- Present comprehensive experiments to demonstrate that the new query types produce meaningful results and the proposed algorithms are efficient and scalable

Thanks

And

Questions?