DASFAA 2013, *Wuhan, China*

Keyword-matched Data Skyline in Peer-to-Peer Systems

Khaled M. Banafaa, Ruixuan Li, *Kunmei Wen,*
Xiwu Gu, and Yuhua Li

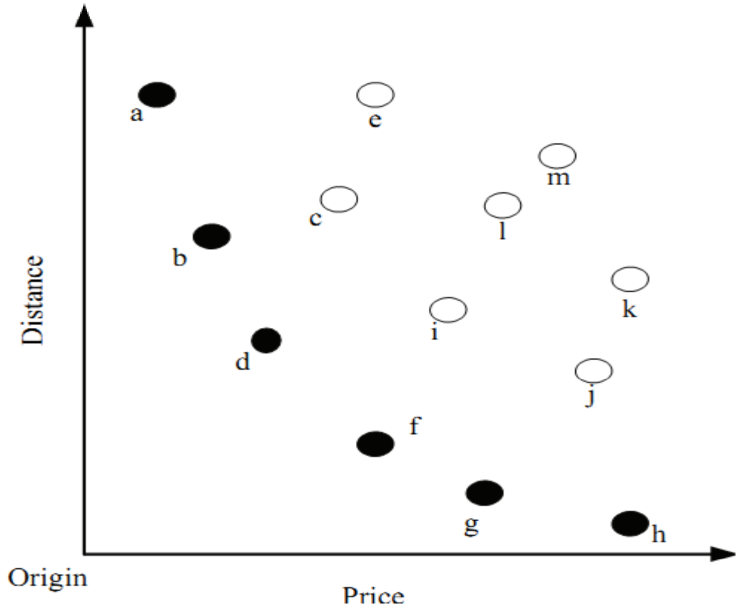
Huazhong University of Science and Technology,
Wuhan, China

Outline

- Background
- Motivation
- Keyword-matched Skyline in P2P
  - Ch-isky algorithm
  - Nk-sky algorithm
  - Ck-sky algorithm
- Performance Evaluation
- Conclusion

Skyline

- A Skyline is the set of all non-dominated tuples.
- A skyline point is a point that is not dominated by any other point in all dimensions.

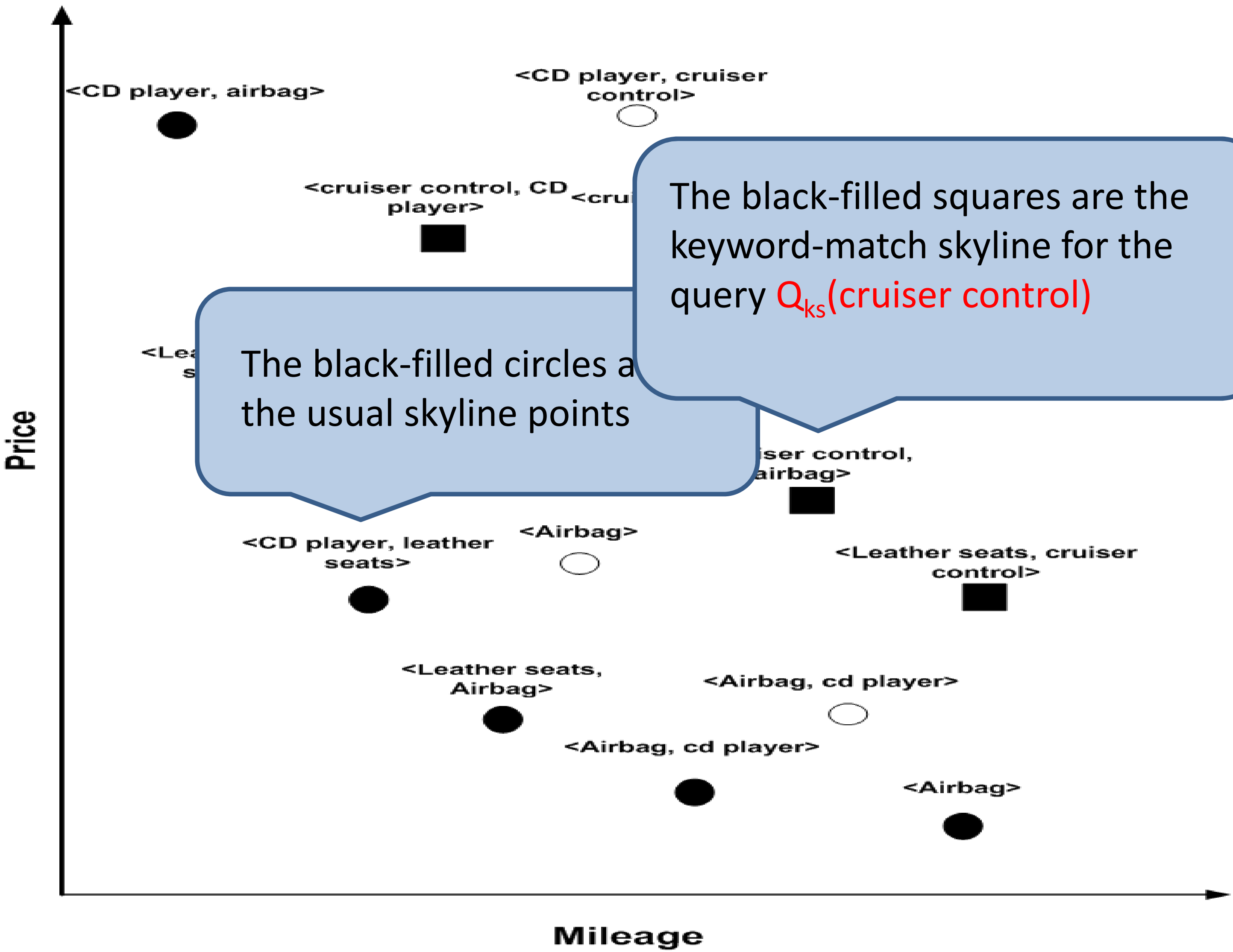


In general, the domination in one dimension is the user preference in that dimension (e.g. cheaper, shorter distance, and lower mileage).

Problems in Skyline

- Traditional skyline may only give the skyline points that are not dominated by any other point in all dimensions.
- However, a user may only be interested in skyline for those points with some features.
- The user preferred features can be represented by some specific keywords.

A Motivating Example



The black-filled circles are the usual skyline points

The black-filled squares are the keyword-match skyline for the query Q_{ks}(cruiser control)

Related Work

- Distributed skyline algorithms do not consider keywords.
  - Feedback-based distributed skyline algorithm (FDS)
  - Distributed SkyLine query (DSL) and SkyFrame
  - Parallel Distributed Skyline (PaDSkyline) and SkyPlan
  - isky: skyline in structured P2P network [ICDCS 2008]
- Keyword-matched skyline in centralized systems using R-tree [Choi, H. et al., Info. Sci. (2013)].
- Traditional keyword search queries in distributed systems and P2P systems ignore skyline incomparability and pruneability features.

Contributions

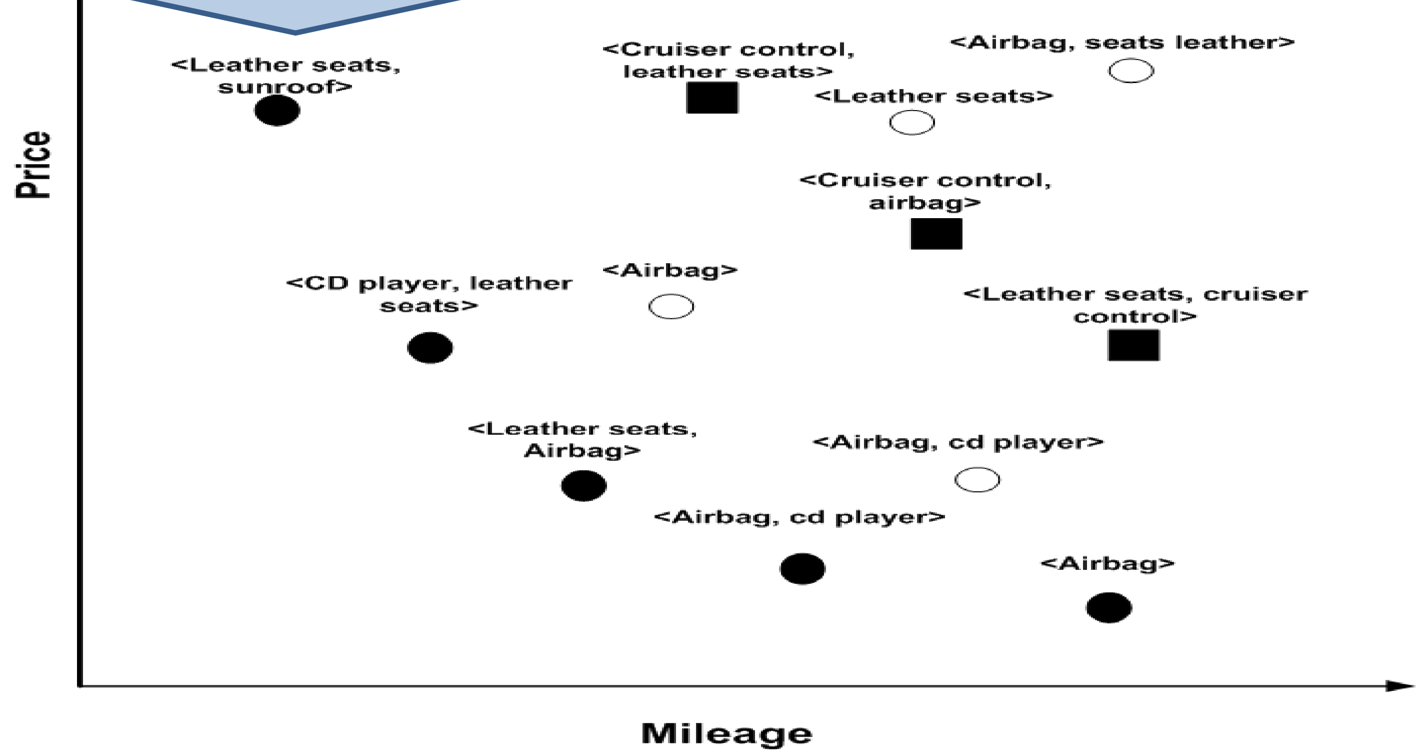
- Bloom filters are used to figure out the candidate peers for query keywords with cover-set tuples and nodes.
- Keyword-matched skyline algorithms are designed and implemented in P2P systems.
- Experiments have been carried out and show that the proposed approaches resulted in reduction of traversed peers while preserving progressiveness.

Problem Definition (I)

A tuple ***t*** in a d-dimensional space D_d is defined as <V,W>

$V = (v_1, v_2, \ldots, v_d)$: a value vector of d-numerical values

$W = (w_1, w_2, \ldots, w_k)$: a set of k keywords for the tuple ***t***.



Problem Definition (II)

- Given a set of query keywords W and a dataset D_d, a keyword-matched skyline query denoted as $Q_{ks}(D_d,W)$, retrieves the set of skyline tuples whose each textual attribute contains all words of W.

$$Q_{ks}(D_d, W) \equiv Q_s(Q_k(D_d, W))$$

Q_{ks}: keyword-matched skyline query

Q_s : skyline query

Q_k: keyword query

Ch-isky

- A modified isky is used to calculate keyword-matched skyline.
- *Filters*
  - *Skyline Filter*

    $$SF_{global} = min_{x \in S}(x_{max}) \quad (3)$$

  - Volume of Dominating Region (VDR)

    $$VDR_p = \prod_{i=1}^{d}(b_i - p_i) \quad (4)$$

    - b_i: Max value in dimension i
    - P_i: Point value in dimension i

Algorithm 1 Ch-isky: Chord-based isky algorithm

```
Input: MinMax-filter, VDR, Querying-Peer, keywords
BEGIN
 if QueryingPeer then
  for each peer P includes an integer (1 to D) in its period do
   send Keyword-matched-Skyline-Query
  end for
 end if
 Calc-key-matched-sky-using-VDR-MinMax()
 Calc-
3, 4 */
Send
if min(
 send-query-to-next-peer(MinMax,VDR)
end if
END
```

> All peers are candidates.
> Keywords are not used to select possible peers which could have the skyline points.

New Approaches

- Data space is partitioned using value attributes.
  - Minimum value preference is assumed.
- Keywords are hashed using Distributed Hash Table (DHT).
- A query undergoes two stages
  - Candidate peers are discovered using Bloom Filters on the keywords.
  - Traversing the peers for skyline in a way that allows pruning and progressiveness.

Node-based Keyword-matched Skyline Algorithm (Nk-sky)

- Setup time: Each peer sends the pair (peer#, keyword) to the responsible peer (using the keyword hash function).
- Query Time: Once a query is triggered, a Bloom filter is used to find out all peers with the query keywords.
  - False positives can be in the results but they do not affect the correctness of the skyline results. (theorem).

False Positives

- False positives are a result of
  1. Bloom Filters: the false positives $p_{f\mathrm{p}}$

$$p_{f\mathrm{p}} = 0.6185^{\mathrm{m/n}}$$

m: Bloom filter bits

n: the number of elements in the set.

  2. A candidate peer may have no point that satisfy query keywords.
     - Because the node keywords are used.
       – (e.g. A peer can have (k1,k2,k3,k4) with points: p1(k1,k3) and p2(k2,k4). A query can have q(k1,k2). No point satisfies the query).

Ck-sky: cover-based keyword-matched skyline algorithm

- Ck-sky comes to minimize the number of false positives.
- In the Setup stage: Only cover-based keyword points (P_{CK}) are hashed.
  - $P_{CK} = \{P_i \in P \mid \forall p_j$ in that peer $p_i.w \not\subset p_j.w\}$

Algorithm 2 Ck-sky: cover-based keyword-matched skyline algorithm

```
Input: MinMax-filter, VDR, Qu
traversed
BEGIN
if QueryingPeer then
  /* first stage */
 Peers-to-be-traversed = get-ca                    Filter()
 for all P ∈ Peers-to-be-traverse
   if p closest above or equal peer to an integer (1 to D) then
     send keywor
   end if
 end for
end if
/* Second stage
Calc-keyword-matc
Calc-VDR-and-Min
Send-results-to-qu
nextPeer = closest-
if min(nextPeer) ⊁ MinMa
  send-query-to-next-peer(MinMax,VDR)
end if
END
```

Each keyword in the query is hashed and the bloom filter for each keyword is sent to

In parallel, the peers are checked using Theorem 1

Skyine Calculations use the filters: VDR to prune points within a peer and MinMax to prune peers.

The query along with the VDR and MinMax are sent to the next candidate peer in a clockwise direction .

Experimental Settings

Table 1. Parameter settings in the experiments

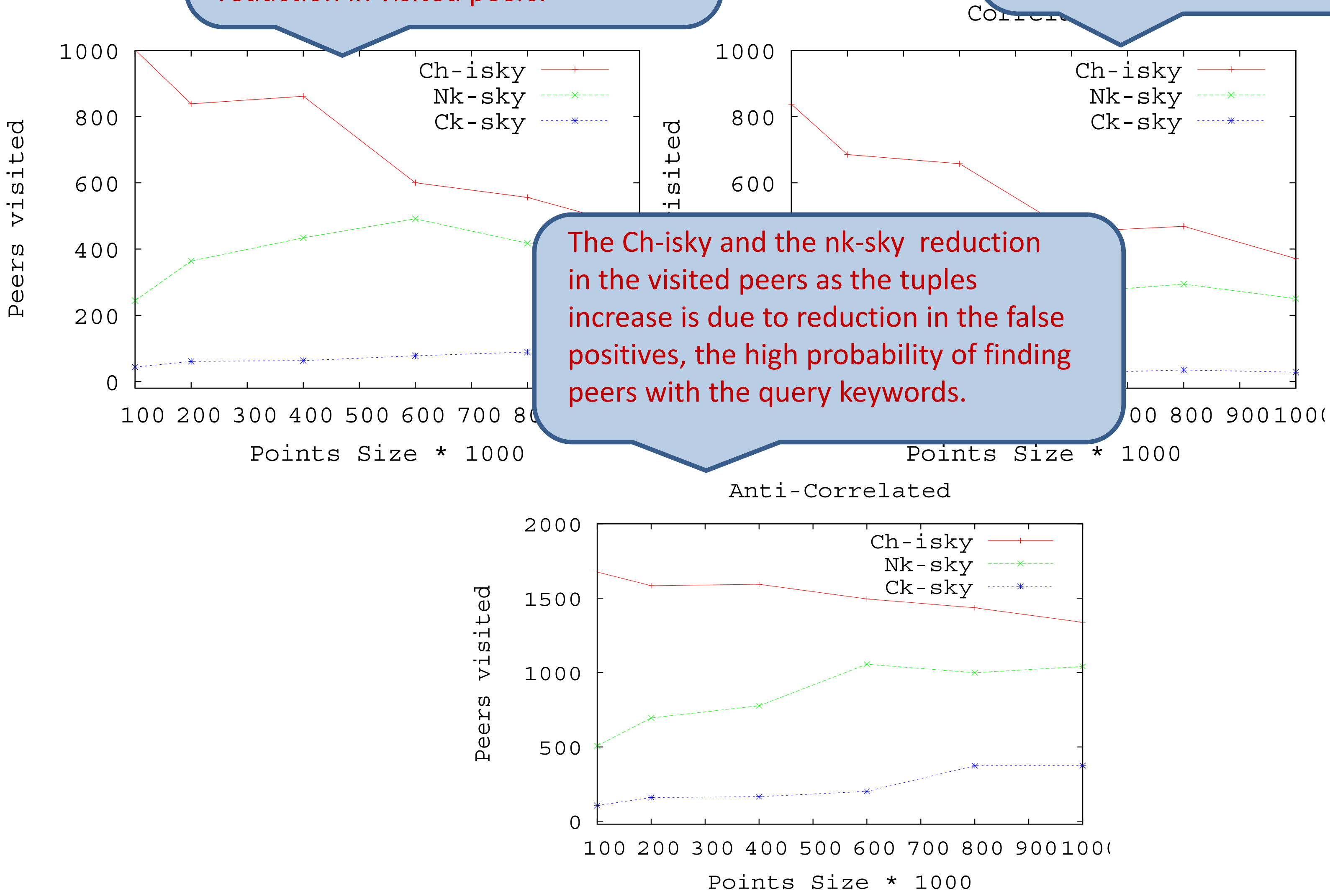
Parameters	**Values**
Cardinality(N) of tuples	100k, 200k, 400k, 600k, 800k, **1M**
Dimensionality	2, **3**, 4, 5
The number of query words (k)	1, **2**, 3, 4, 5
Zipf skew factor (θ)	0.0, 0.2, 0.4, 0.6, **0.8**, 1.0
Distribution(for values)	independent, correlated, anti-correlated
Tuple's keywords	6
Network size (no. of peers)	100, 1000, **2000**, 3000, 4000

Visit ... umbe

As we can see, the nk-sky reduces the number of visited peers. The Ck-sky reduces the number of false positives resulting in more reduction in visited peers.

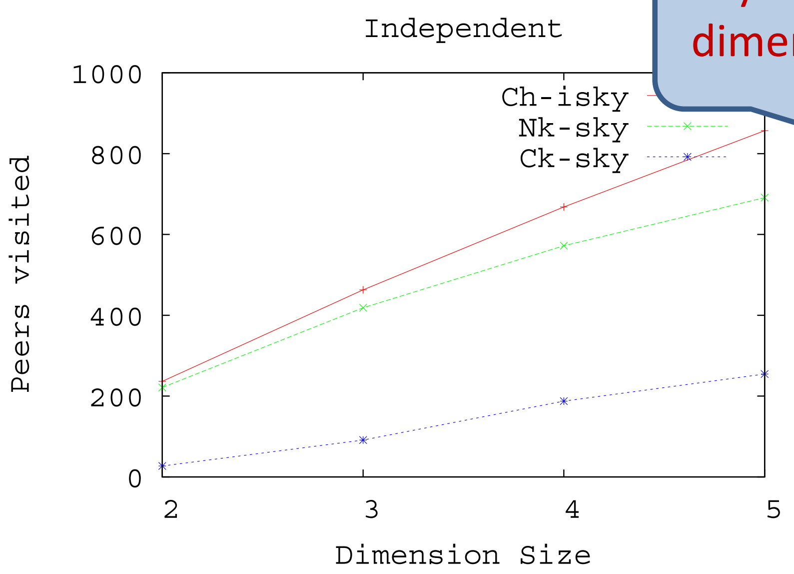
In the Correlated data sets, the visited peers are less than the other because less number of skyline points are expected. This also affects other measures as we see later.

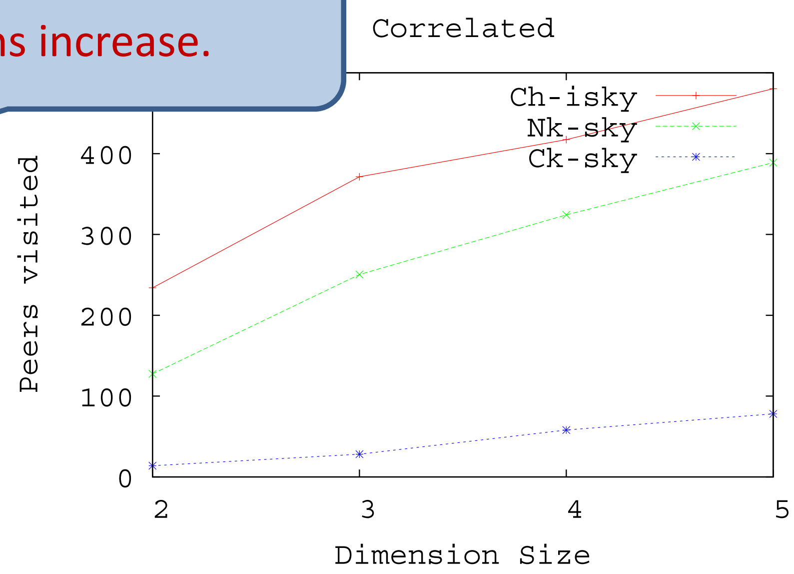
The Ch-isky and the nk-sky reduction in the visited peers as the tuples increase is due to reduction in the false positives, the high probability of finding peers with the query keywords.

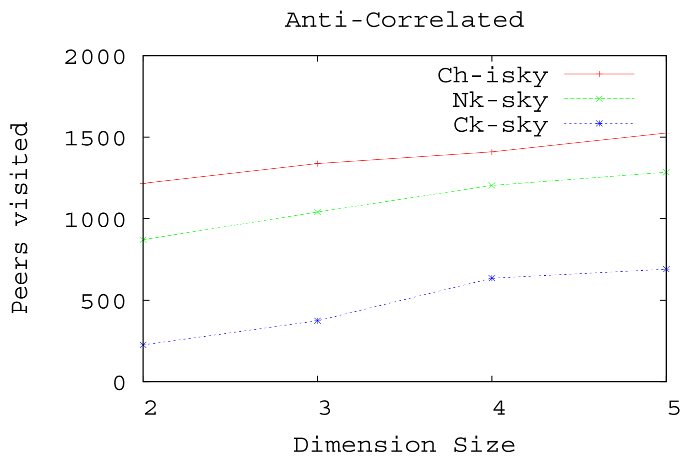


Visited Peers vs. Dimensions

Skyline points increase as dimensions increase.

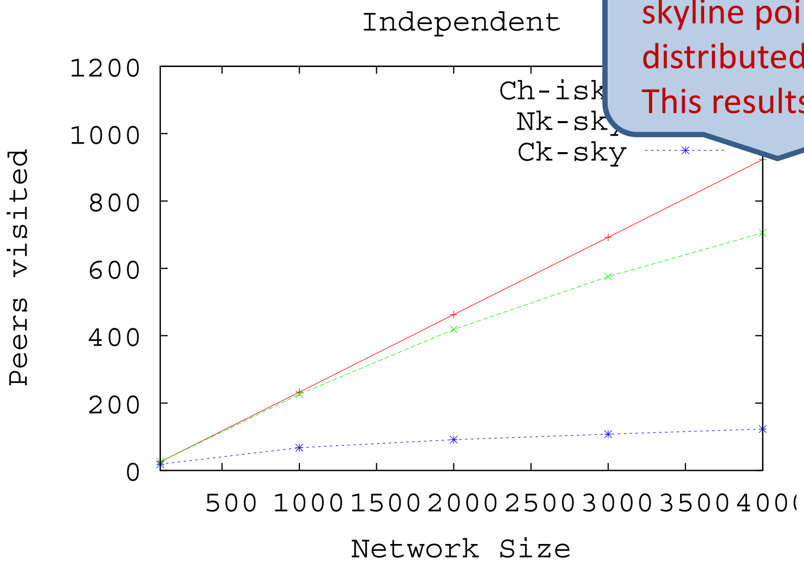


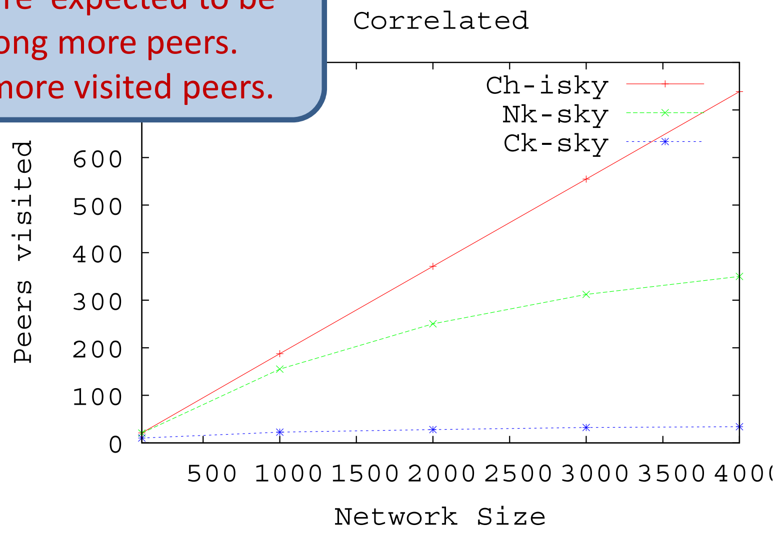


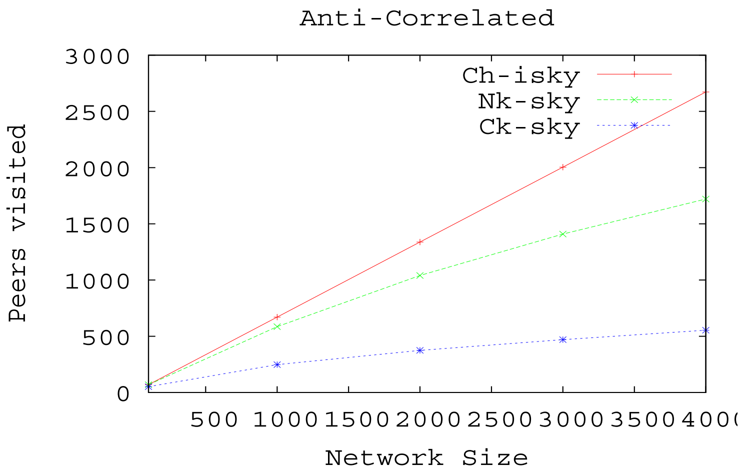


Traversed Peers vs. Network Size

As network size increases, the skyline points are expected to be distributed among more peers. This results in more visited peers.

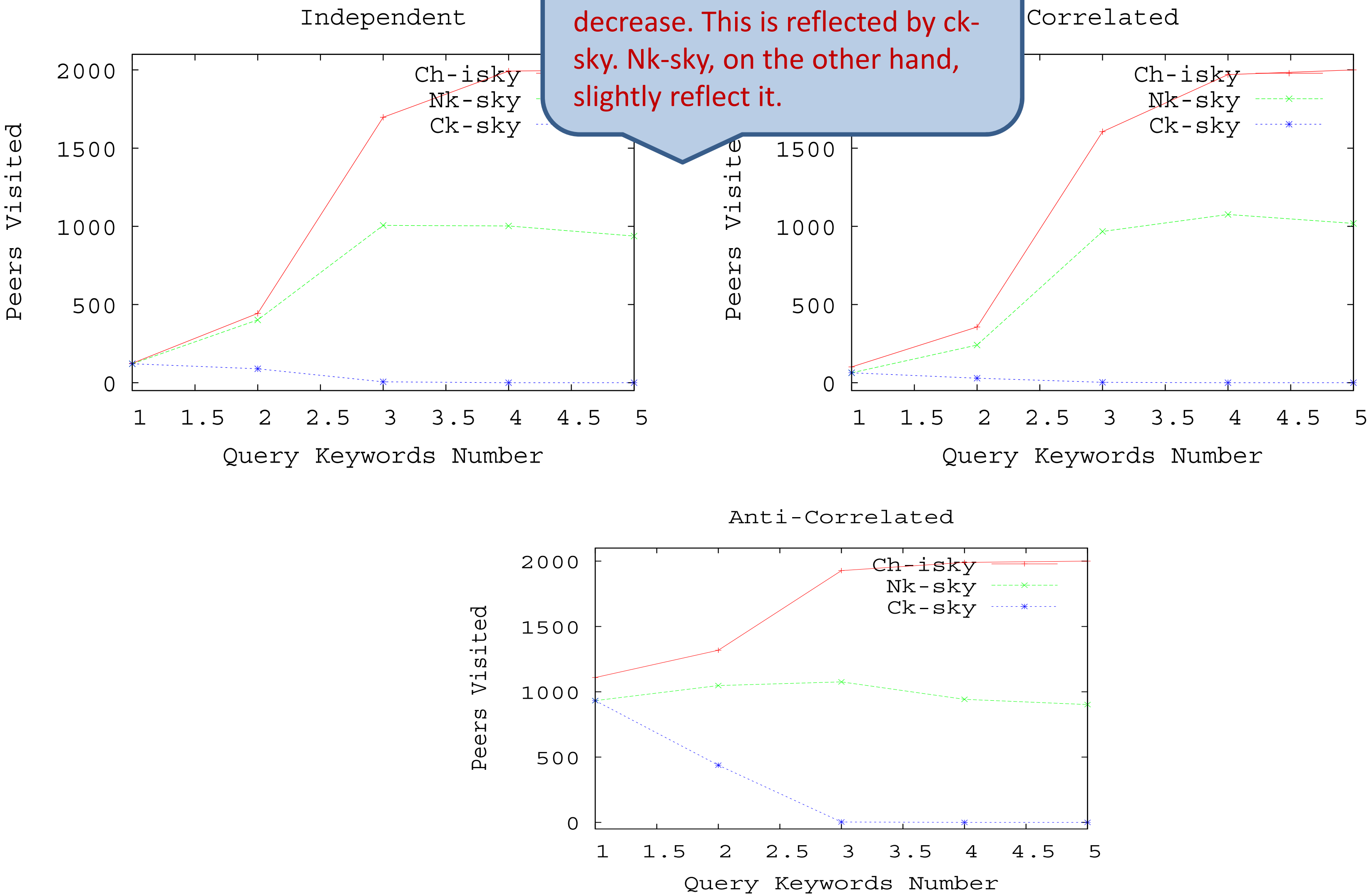






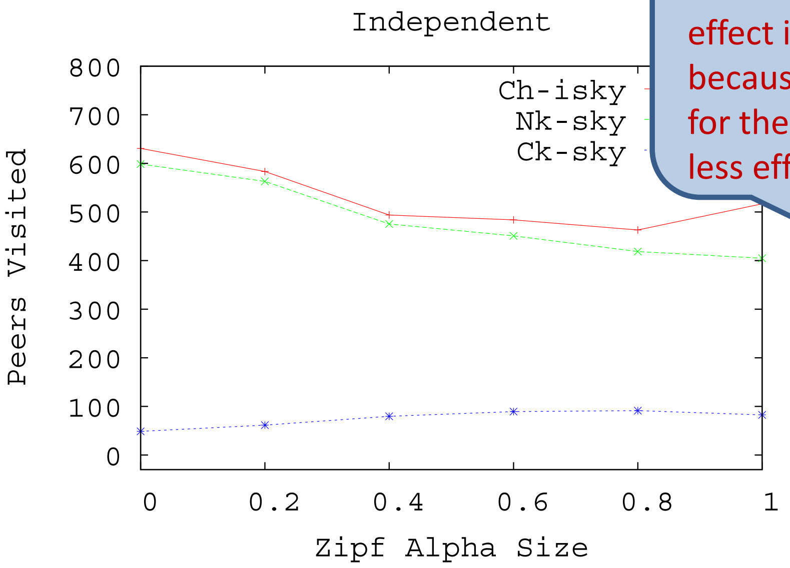
Traversed Peers vs. Query Keyword

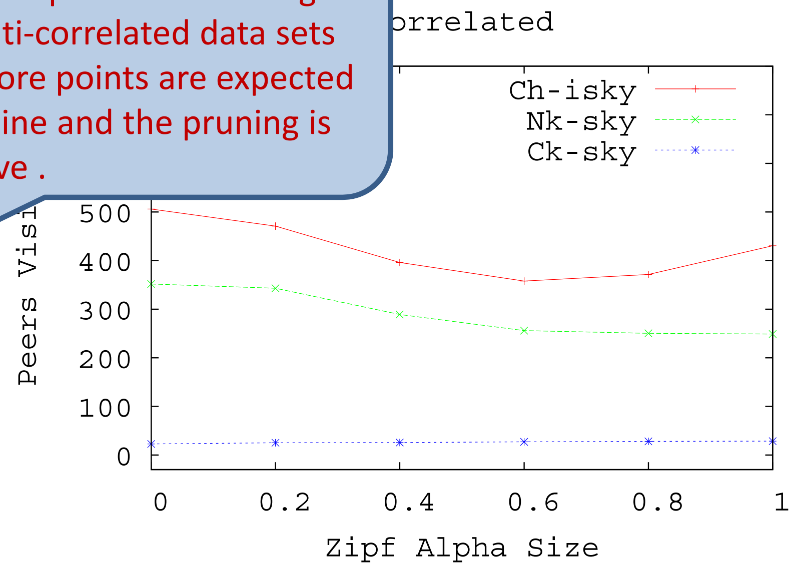
As query keywords increase, the number of positive peers decrease. This is reflected by ck-sky. Nk-sky, on the other hand, slightly reflect it.

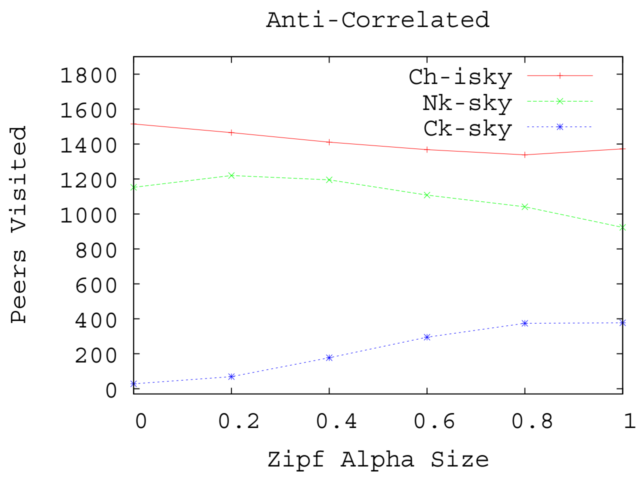


Traversed Peers vs. Skewness

Zipf does not have a very big effect on the visited peers. It has a slight effect in anti-correlated data sets because more points are expected for the skyline and the pruning is less effective .







Conclusions

- This paper addresses keyword-matched skyline in peer-to-peer systems.
- Node and tuple-based algorithms (Nk-Sky) are designed to solve keyword-matched skyline in P2P systems.
  - The algorithms use DHTs functions and Bloom filters to minimize the number of traversed peers.
- Ck-sky, a cover-based keyword-matched skyline algorithm, can greatly reduce false positives peers resulted from Bloom filters.

Thanks for your attention

Contact information:

- Ruixuan Li
- Huazhong University of Science and Technology
- rxli@hust.edu.cn
- http://idc.hust.edu.cn/~rxli/