The V*-Diagram: A Query-Dependent Approach to Moving KNN Queries

Sarana Nutanong, Rui Zhang, Egemen Tanin, Lars Kulik

Dept. of Computer Science and Software Engineering
University of Melbourne

Motivation

Consider two scenarios:

- a driver in a GPS-equipped car finding the nearest gas station along the route of a trip;
- an ambulance maintaining a list of k nearest hospitals while driving around a city.

These scenarios are examples of *moving k nearest neighbor queries (MkNN)*.

Simple Approach

The Voronoi Diagram

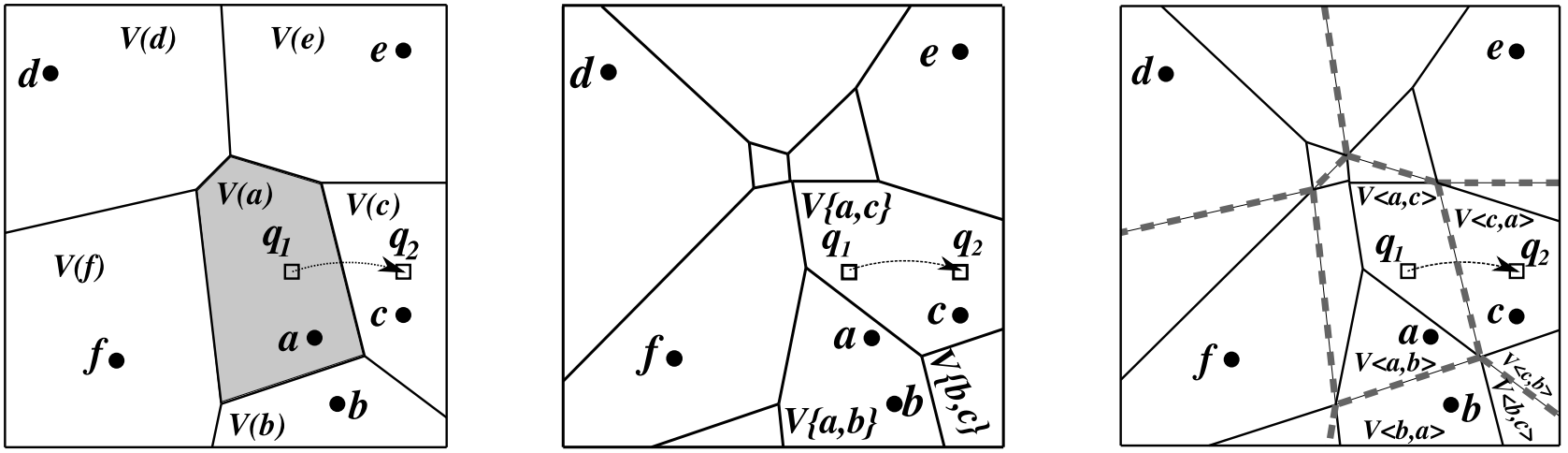


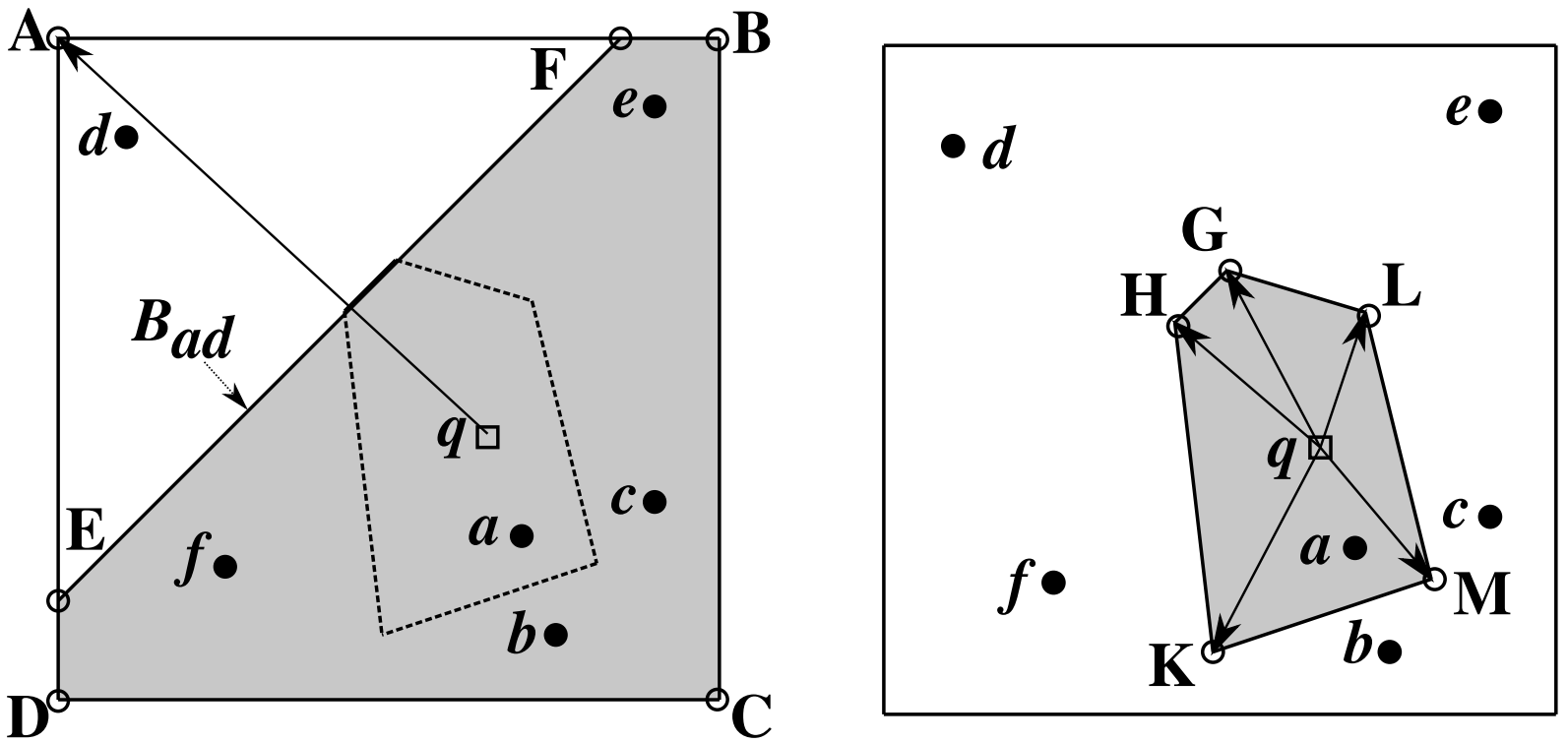
Figure 1: Voronoi diagrams

Drawbacks:

1. Expensive precomputations (quadratic wrt k [Aurenhammer and Schwarzkopf])
2. Inefficient update operations
3. No support for dynamically changing k values

Best Existing Approach

Influence-set Retrieval [Zhang et al., 2003]



(a) Bisector B_{ad} is discovered as a boundary.

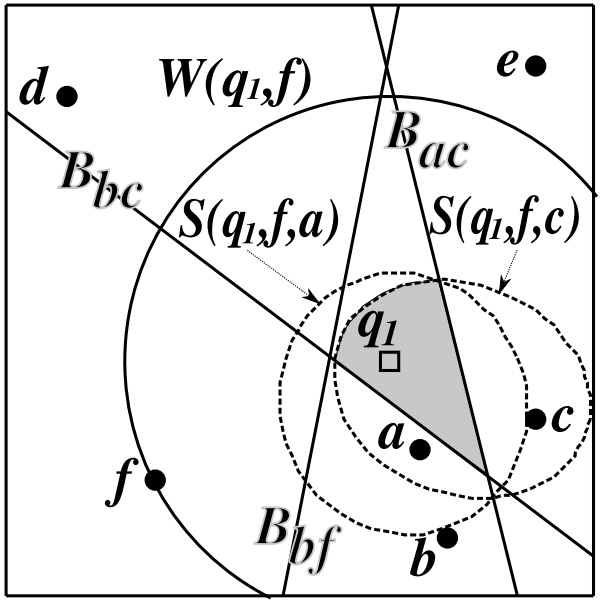
(b) All boundaries are discovered

Figure 2: Computing a Voronoi cell locally

Our Approach: V*-Diagram

Objectives:

1. Requires *no precomputation*
2. Supports *insertions* and *deletions* of objects
3. Handles *dynamically changing* k

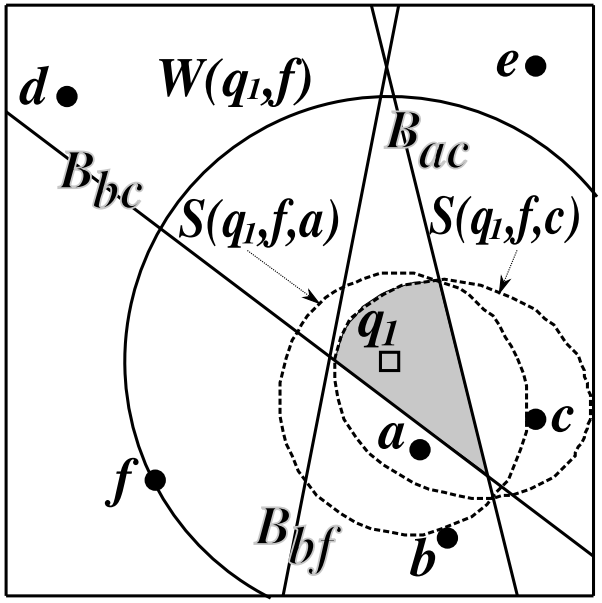


Our Approach: V*-Diagram

Objectives:

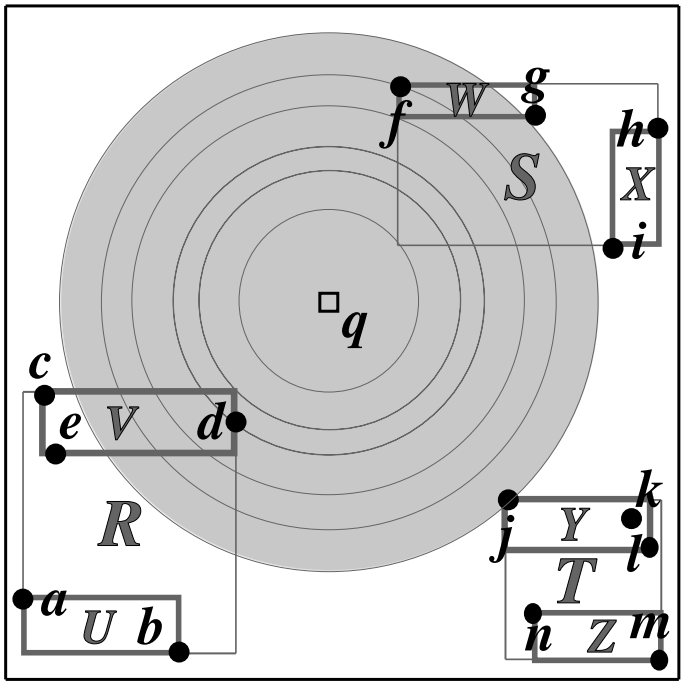
1. Requires *no precomputation*
2. Supports *insertions* and *deletions* of objects
3. Handles *dynamically changing* k

Result: Outperforms the best practice [Zhang et al.] by *2 orders of magnitude*



The V*-Diagram

Known Region



If the known NNs to q are

$$\{d, f, j\},$$

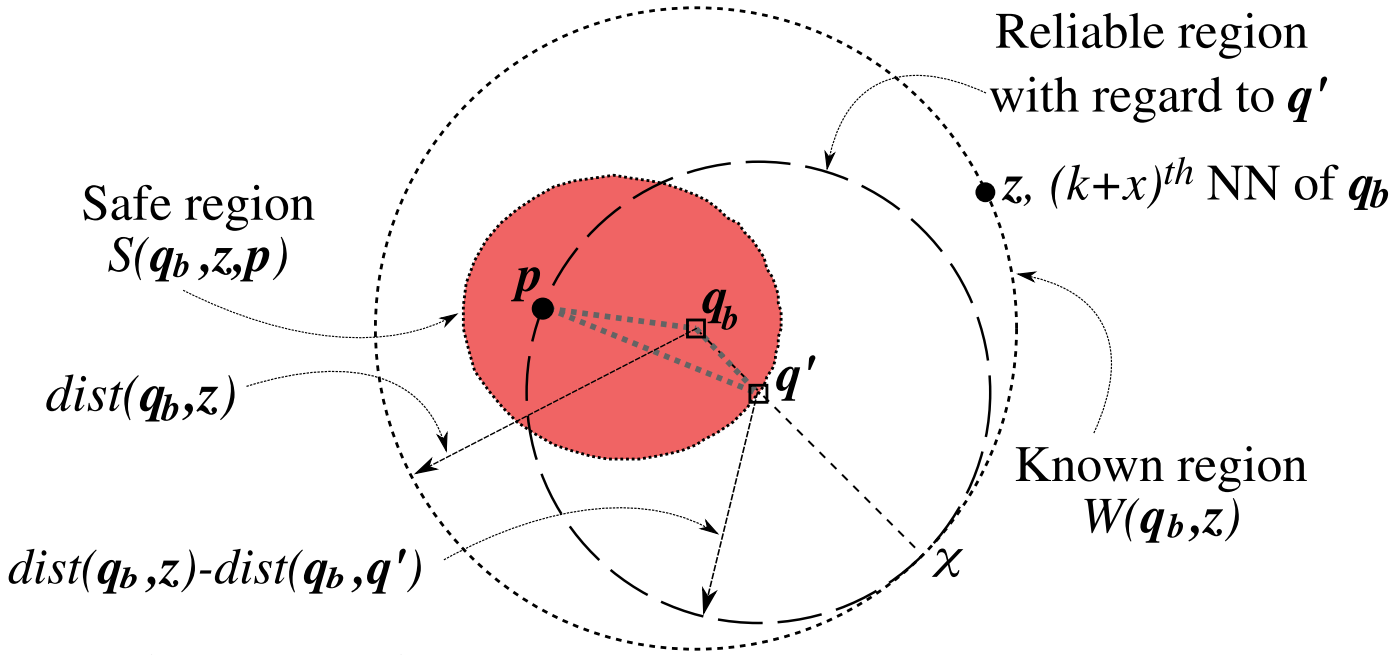
the know region $W(q, j)$ is

$$\{v : dist(q, v) \leq dist(q, j)\}.$$

The V*-Diagram

Safe region wrt a data point

We retrieve $(k + x)$ objects. In this example, k and x are 1, so we retrieve $\boldsymbol{p}$ and $\boldsymbol{z}$.



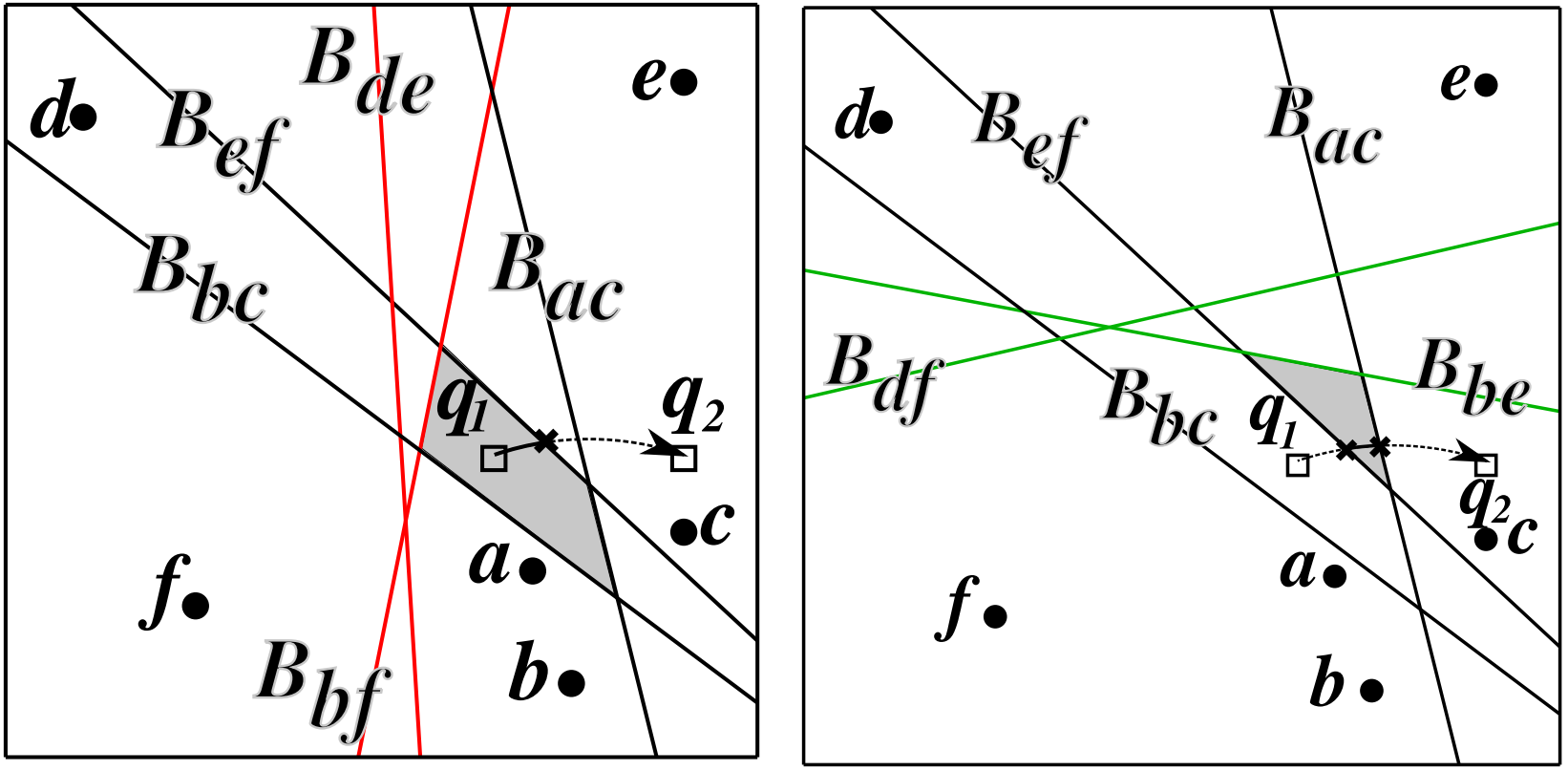
If $\boldsymbol{q'} \in S(\boldsymbol{q_b}, \boldsymbol{z}, \boldsymbol{p})$ then,

$$\forall \boldsymbol{p'} \notin W(\boldsymbol{q_b}, \boldsymbol{z}), dist(\boldsymbol{q'}, \boldsymbol{p}) < dist(\boldsymbol{q'}, \boldsymbol{p'}).$$

$$S(\boldsymbol{q_b}, \boldsymbol{z}, \boldsymbol{p}) = \{\boldsymbol{q'} : dist(\boldsymbol{p}, \boldsymbol{q'}) \leq dist(\boldsymbol{q_b}, \boldsymbol{z}) - dist(\boldsymbol{q_b}, \boldsymbol{q'})\}$$

The V*-Diagram

The Fixed-rank Region (FRR) [Kulik and Tanin, 2006]



(a) $\langle a, c, b, f, e, d \rangle$ (b) $\langle a, c, b, e, f, d \rangle$

Figure 3: Incremental rank update

The V*-Diagram

Integrated Safe Region (ISR) and V*-kNN

ISR is an intersection of

1. the safe region wrt k^{th} NN, $S(\boldsymbol{q_b}, \boldsymbol{z}, \boldsymbol{p_k})$;
2. the FRR of the $(k+x)$ NNs of $\boldsymbol{q_b}$.

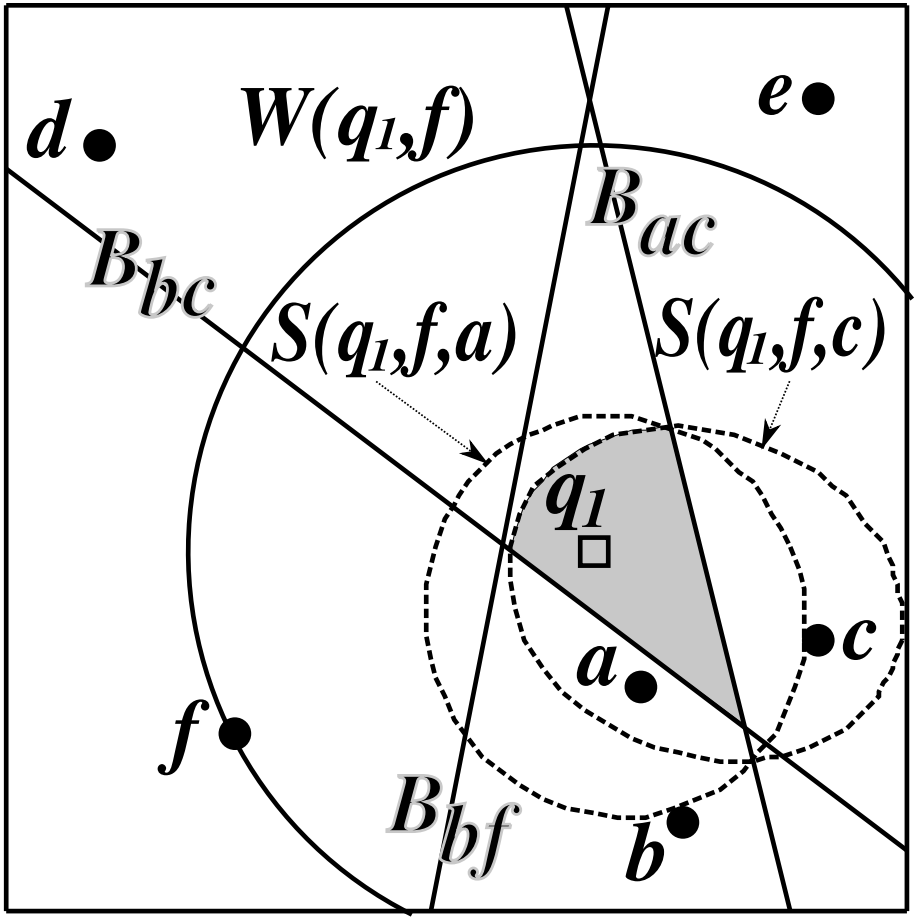
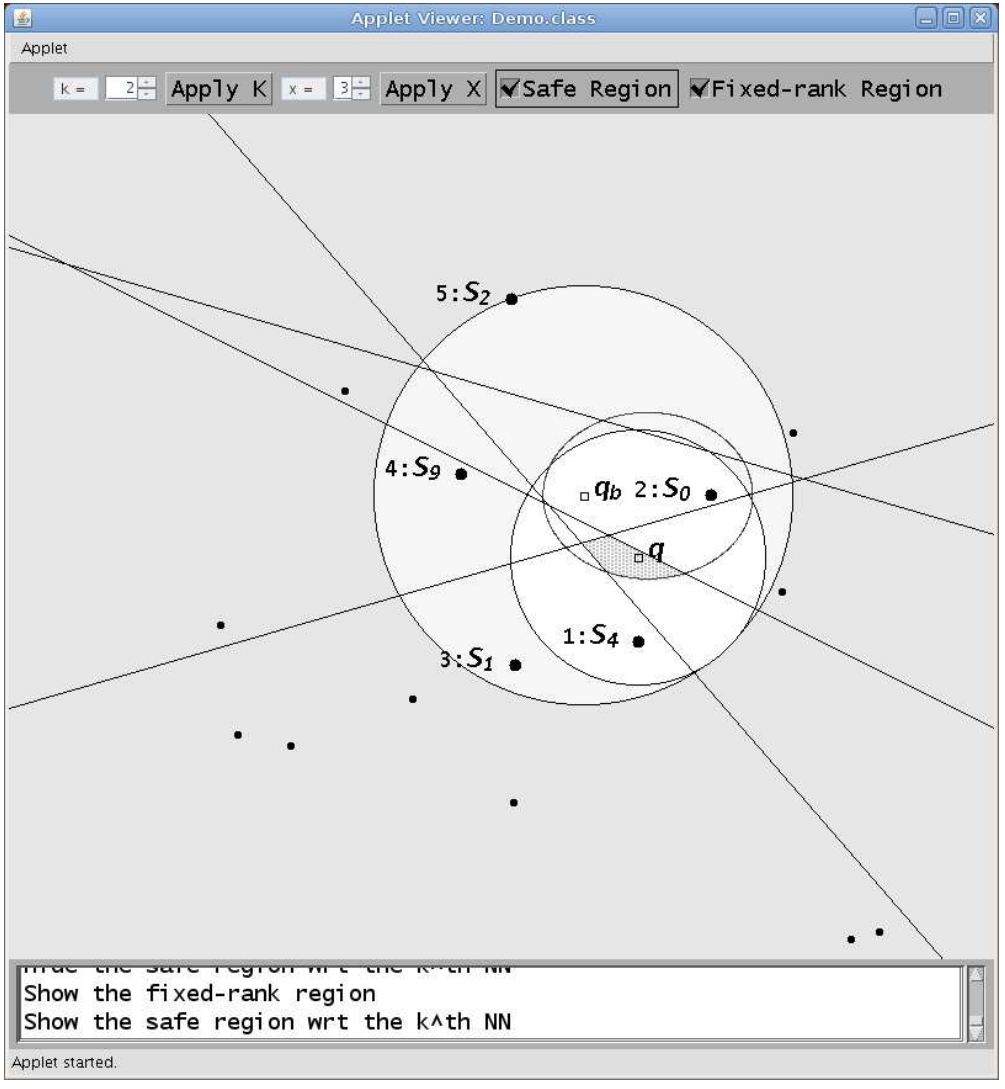


Figure 4: V*-kNN Example ($k = 2$, $x = 2$)

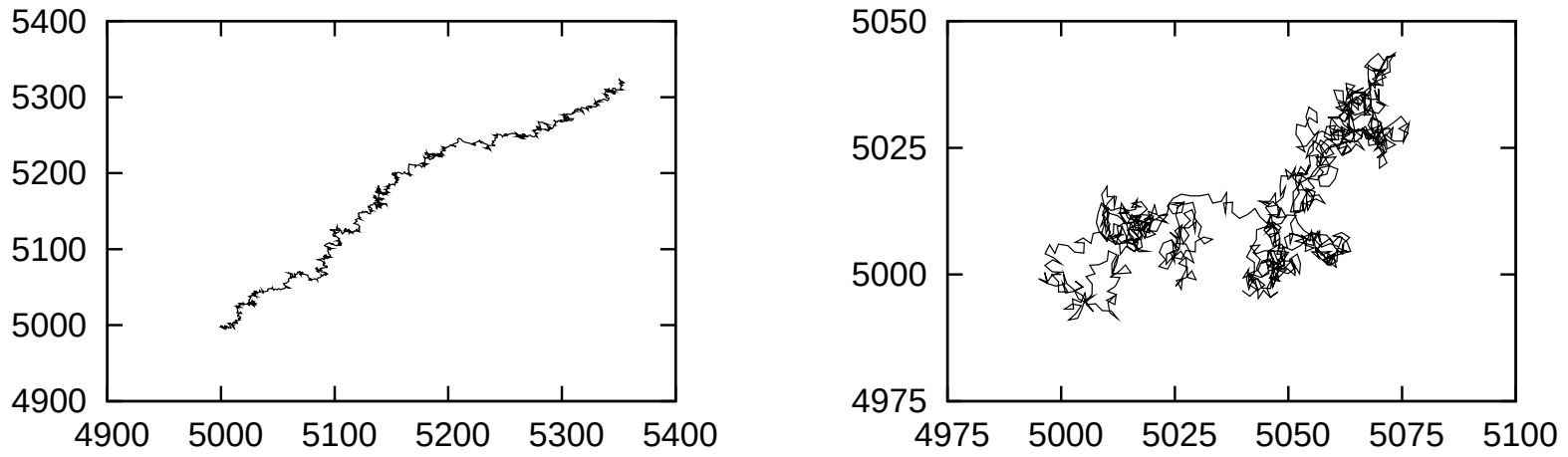
V*-kNN Algorithm

http://www.csse.unimelb.edu.au/~sarana/demo.html



Experiments

- Data Structure: R*-trees (1-kB block size)
- Comparative Method: RIS-kNN [Zhang et al.]
- Trajectories:



(a) Directional (D) (b) Random (R)

Figure 5: Trajectory types

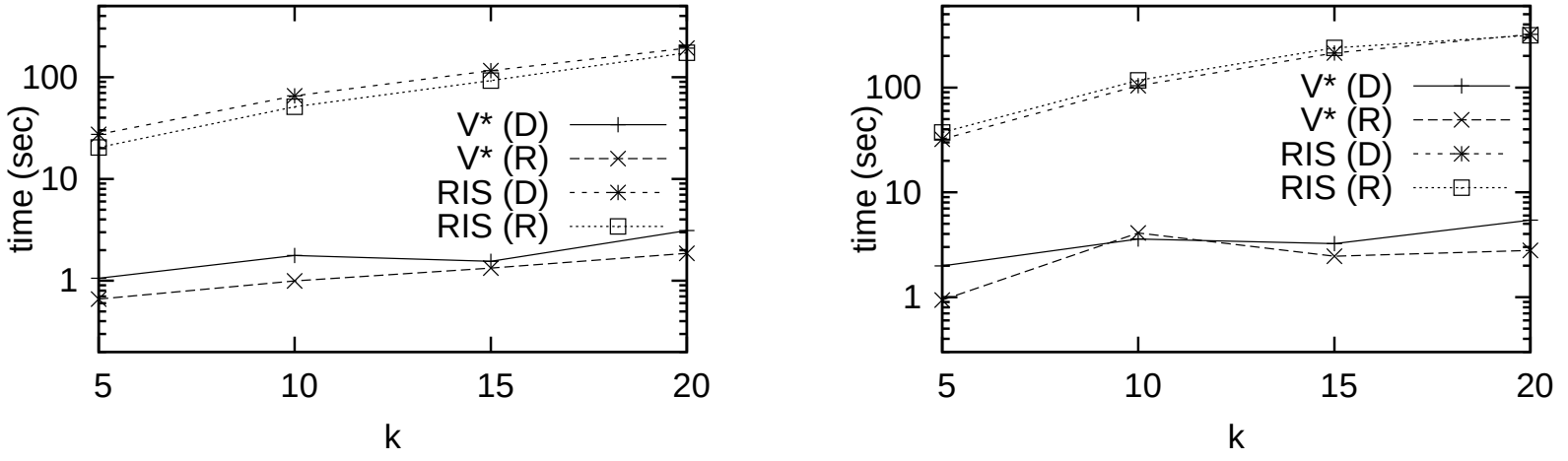
Experiments

total cost wrt k

The two datasets are:

(a) $65,743$ postal addresses from California

(b) $119,897$ postal addresses from North-Eastern USA



(a) Total Cost (California)

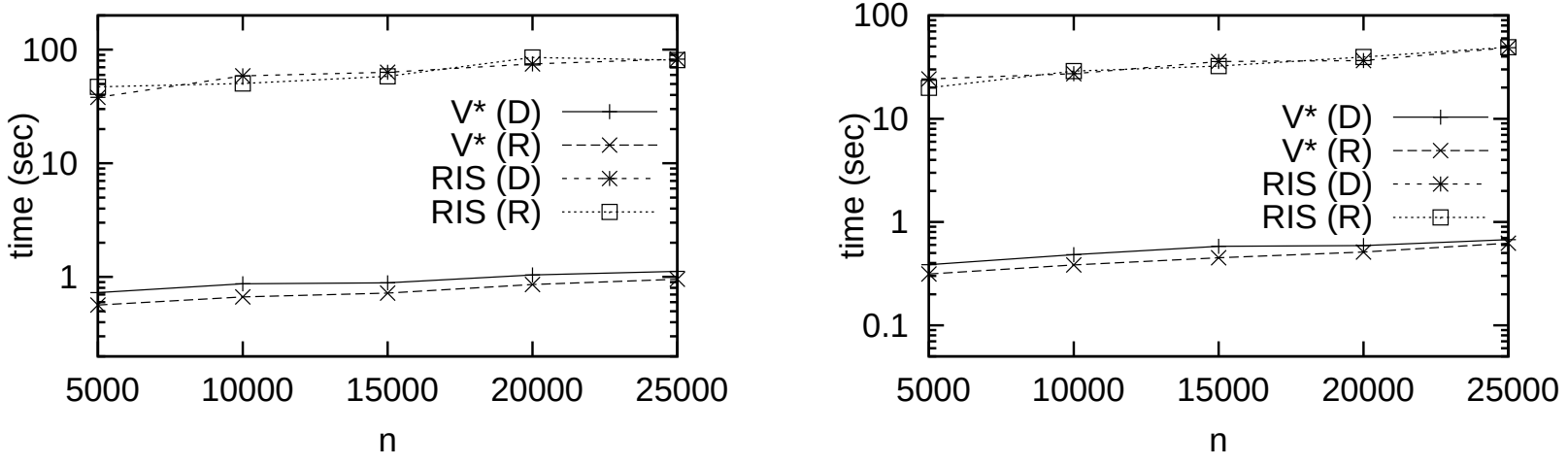
(b) Total Cost (North-Eastern USA)

Figure 6: Effect of k

Experiments

total cost wrt n

We use two types of distributions, uniform and Zipfian.



(a) Total Cost (Uniform)

(b) Total Cost (Zipfian)

Figure 7: Effect of dataset size

Conclusions

- The V*-Diagram constructs a safe region using:
  1. the location of the query point,
  2. kNN-search coverage (known region),
  3. known data points.
- V*-kNN is *local*, *incremental* and *dynamic*.
- V*-kNN outperforms the best existing technique by two orders of magnitude.

Continuing Work

The V*-Diagram in a spatial network

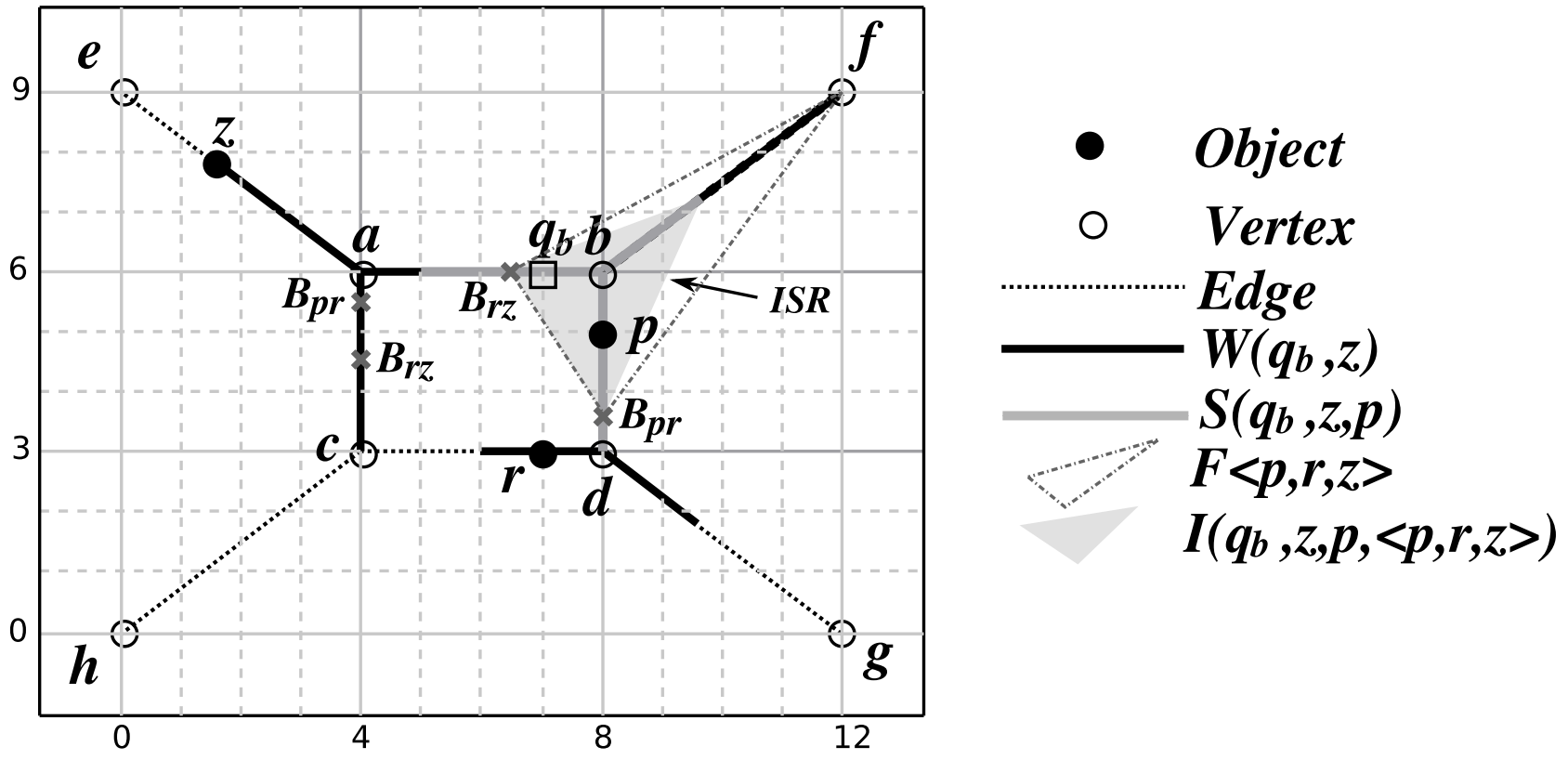


Figure 8: The V*-Diagram in a spatial network ($k = 1$ and $x = 2$)

Acknowledgments

This work is supported under

- the Australian Research Council's Discovery funding scheme (project number DP0880215);
- NICTA Victoria Laboratory.