Clustering-Based Graphs for Large Scale Approximate Nearest Neighbor Search

Javier Alvaro Vargas Muñoz

Advisor: Prof. Dr. Ricardo da Silva Torres
Co-Advisor: Prof. Dr. Zanoni Dias

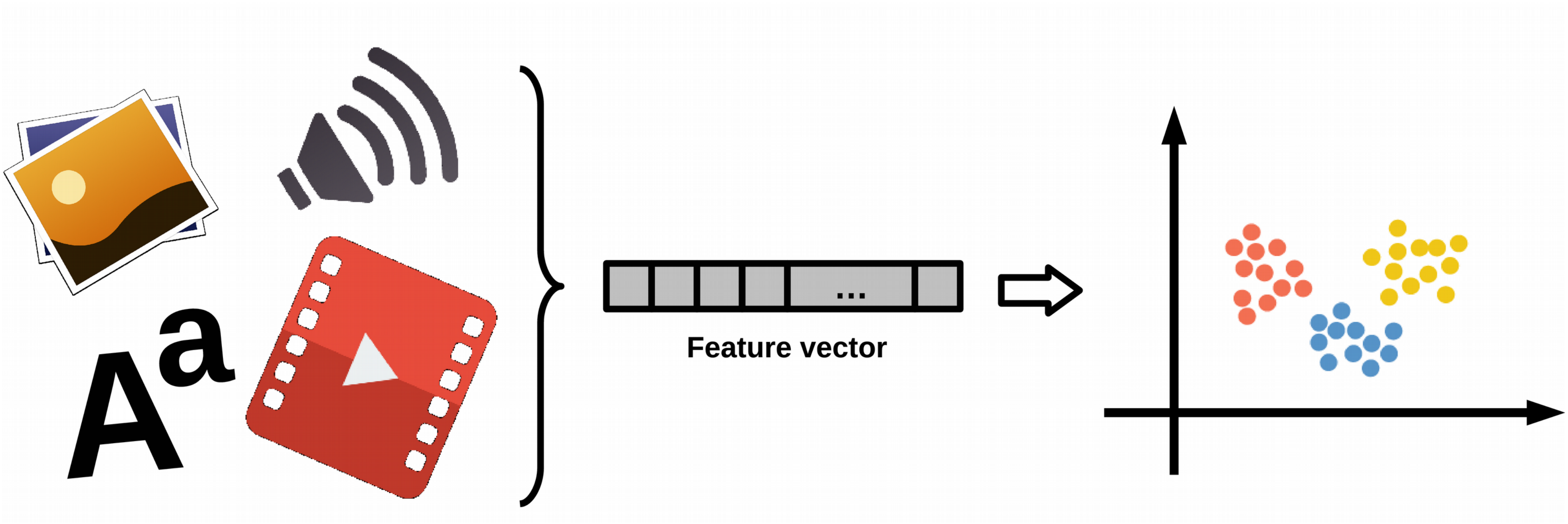
Institute of Computing – University of Campinas

November 28, 2017

Outline

1. Introduction
2. Methodology
3. Results
4. Schedule

Introduction



Introduction

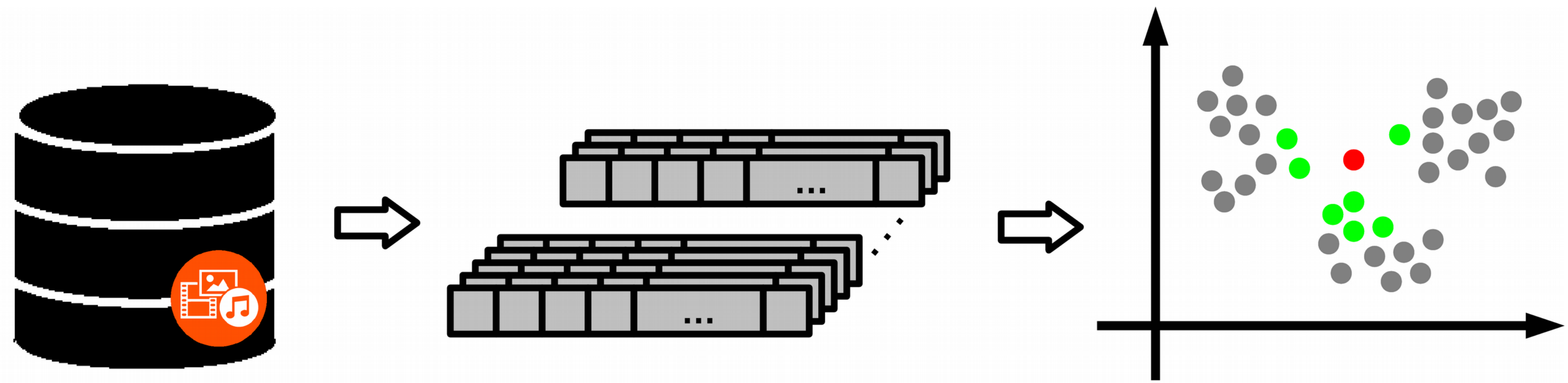
- Computer Vision
  - Find the best matches for local image features
  - Global image feature matching for scene recognition
  - Matching deformable shapes for object recognition
- Machine learning
- Multimedia retrieval

Introduction

- K nearest neighbors search problem

$Given\ a\ set\ P=\{p_1, p_2, p_3, \ldots, p_N\},\quad P \subset \mathbb{R}^d,$
$a\ distance\ function\quad D:\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R},\quad a\ query\ point\quad q \in \mathbb{R}^d$

$$KNN(q, K, P) = A,\ \ where\ A \subseteq P\ \ \wedge\ \ |A| = K\ \ \wedge\ \ \forall x \in A, y \in P - A,\ \ D(q,x) < D(q,y)$$



Nearest neighbor search

Introduction

- Exact methods
  - Naive approach: linear search (brute force)
    - Impractical in large datasets
  - Data structures:
    - KD-Tree, BallTree
    - Search in logarithmic time in low dimensional data
    - Quickly converge to linear search as dimensionality increases

Introduction

- Approximate methods (ANNS)
  - Fast search, with small loss in precision
  - Three classic approaches
    - Tree partitioning structures
    - Hashing-based techniques
    - **Nearest neigbors graphs**

Introduction

- Tree partitioning structures
  - Index contruction: at each tree level, objects are split into subsets, based on some criteria
  - Search: traversing from the root to the leaves, using the same criteria for split
  - Examples:
    - FLANN library: Randomized KD-Trees, Hierarchical k-means tree, Hierarchical Clustering Tree
    - PCA-Tree, RP-Tree, Cover-Tree, VP-Tree

Introduction

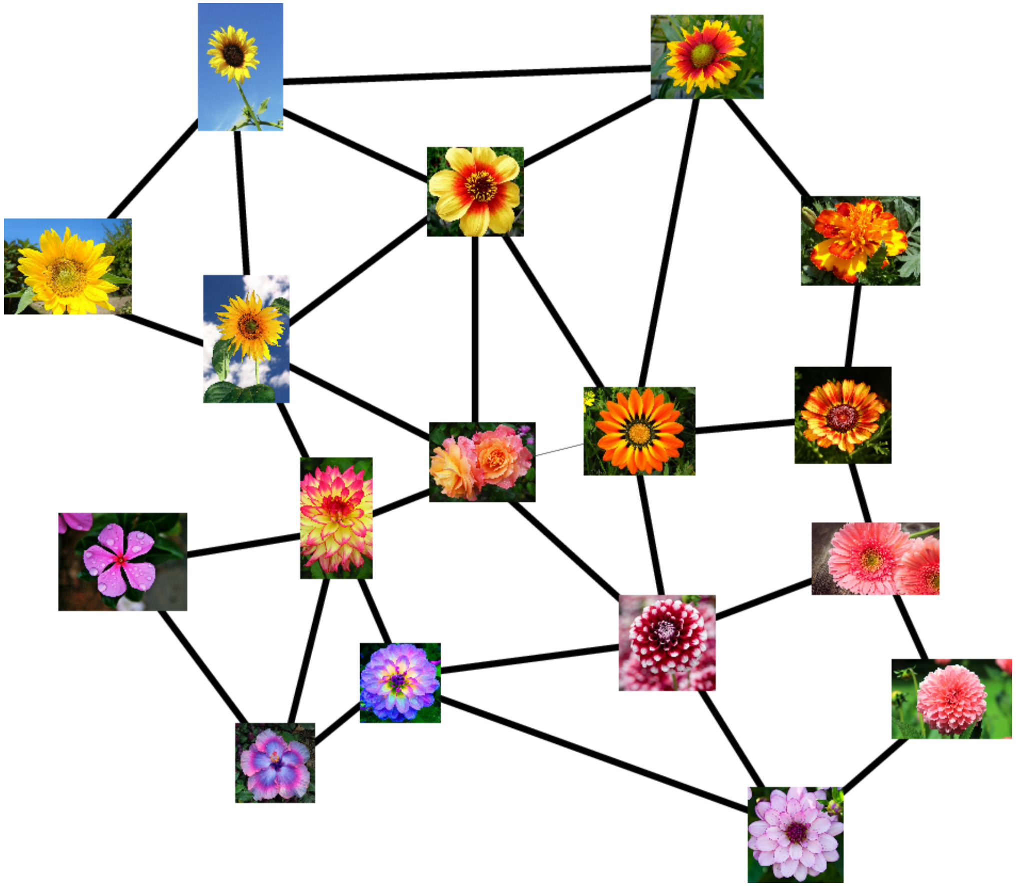
- Hashing-based techniques
  - Index contruction: map similar objects to near positions in the hash tables (buckets)
  - Search: hash the query object into a bucket, and uses the data objects in the bucket as the candidate set of the results
  - Examples: Locality Sensitive Hashing (LSH), Multi-Probe LSH

Introduction

- Nearest neighbors graphs (NN graphs)
  - Graph construction: link each object to the most similar ones
  - Search: start in some (random) vertex and, in a greedy maner, traverse the graph towards the closest points to the query, until some stopping criterion is reached
  - Recents works and benchmarks have shown considerable gains over other approaches
  - Examples: FANNG, HNSW, SW, KGraph

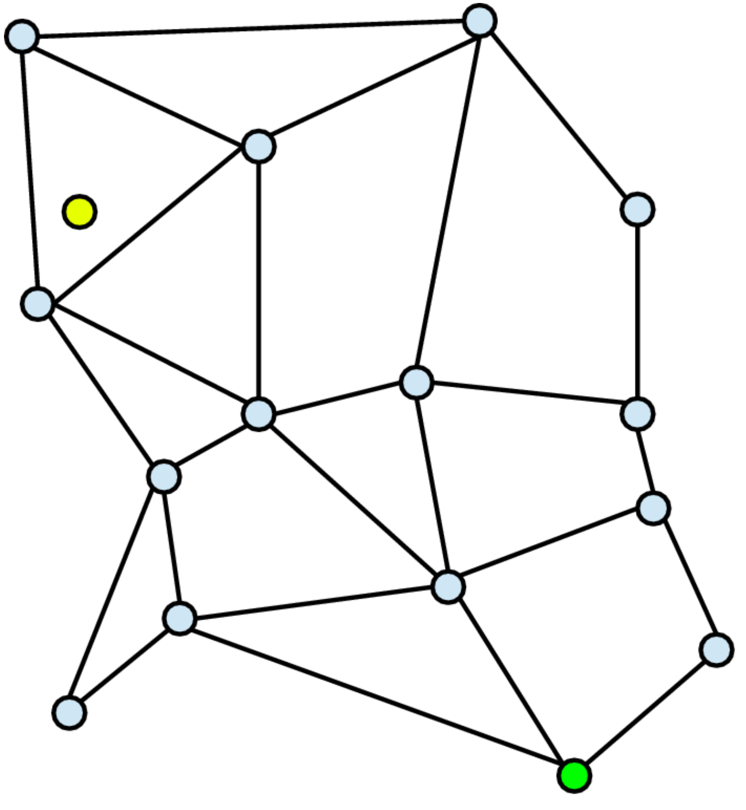
Introduction

- Graph example:



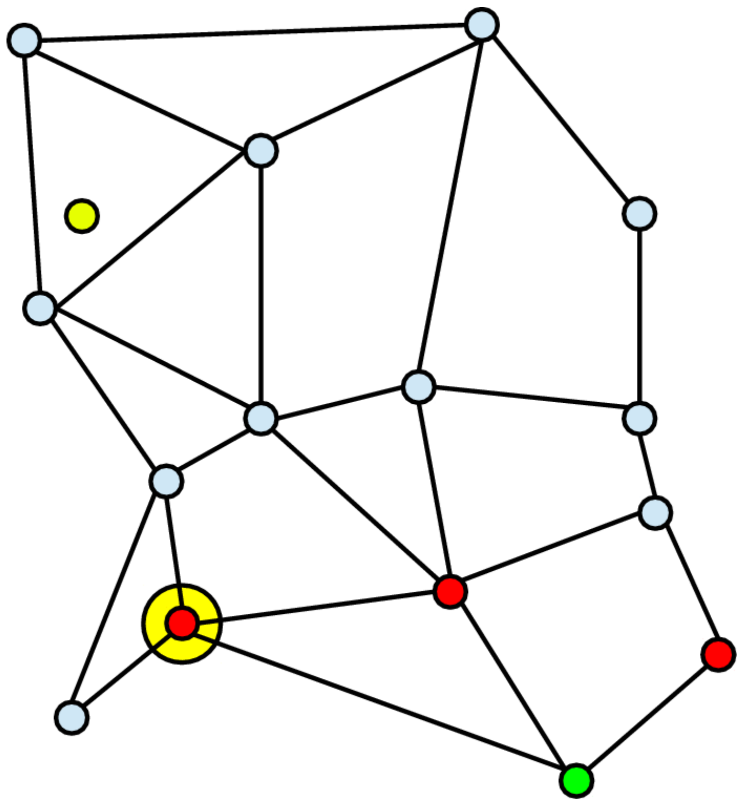
Introduction

- Search example (greedy algorithm)



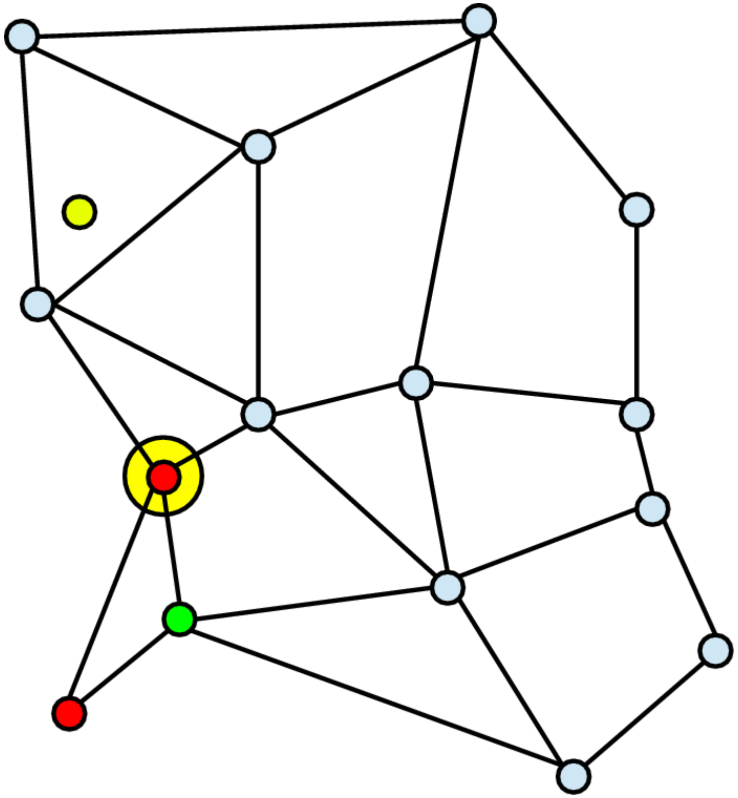
Introduction

- Search example (greedy algorithm)



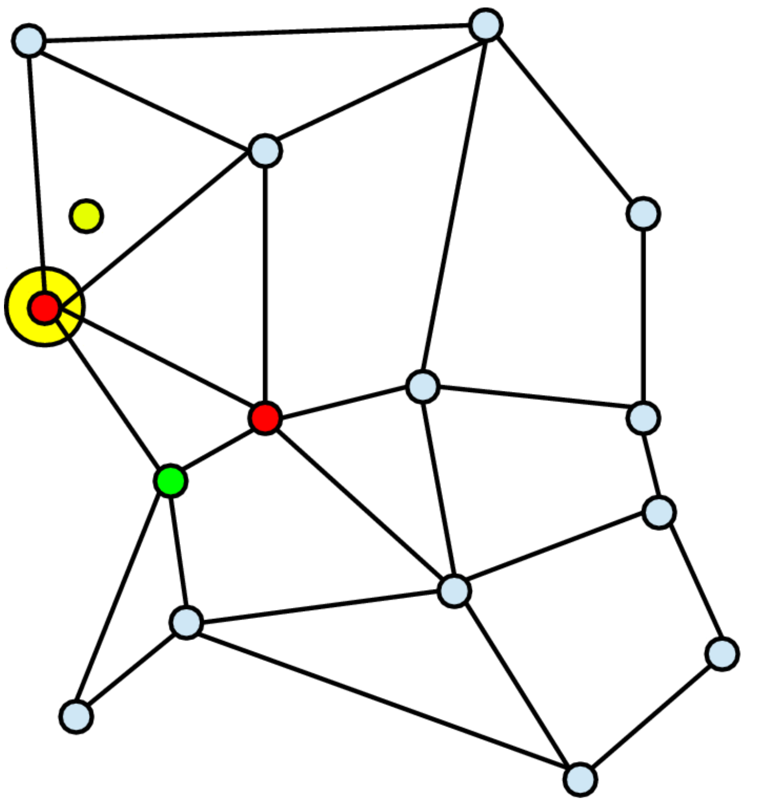
Introduction

- Search example (greedy algorithm)



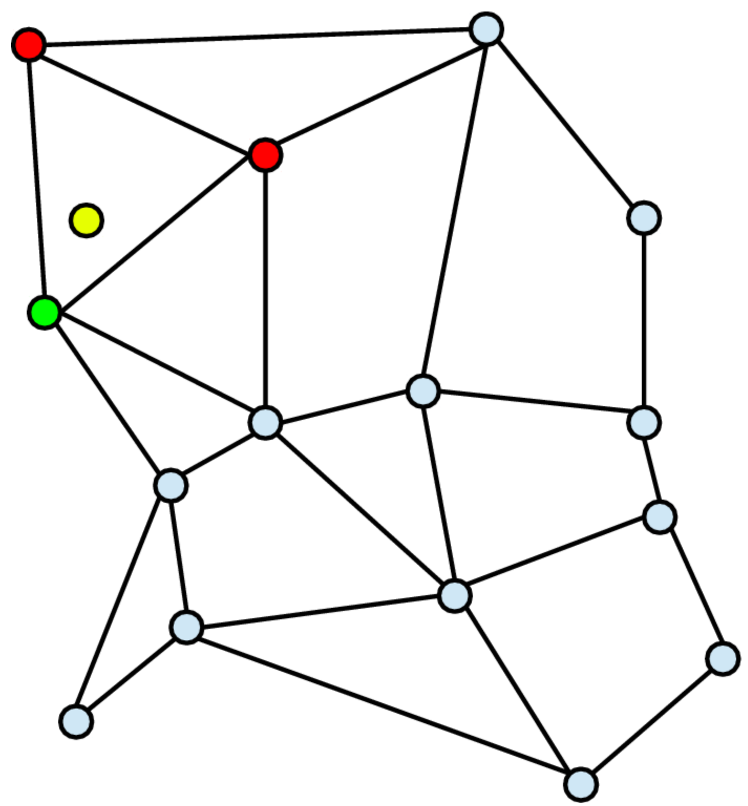
Introduction

- Search example (greedy algorithm)



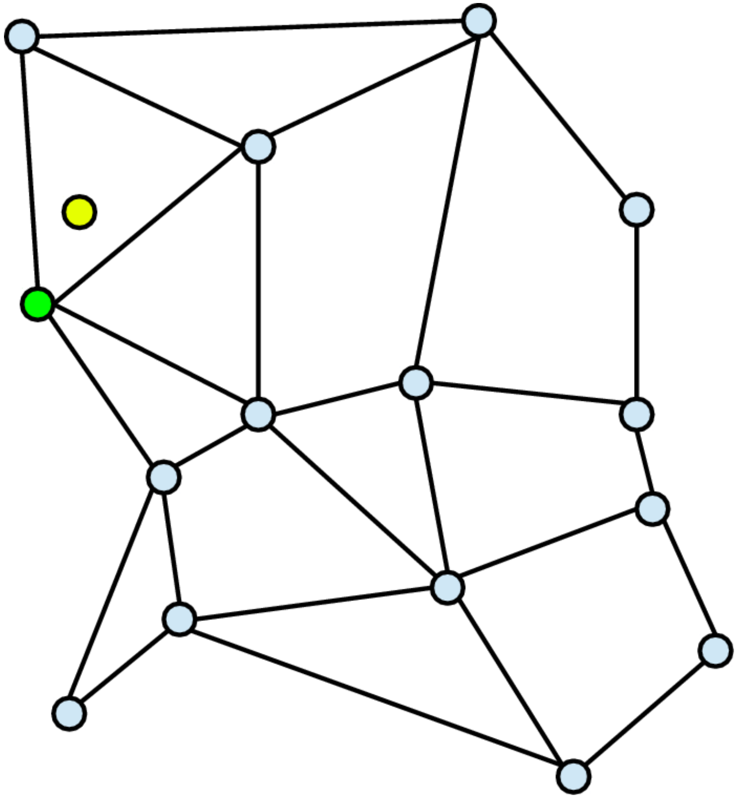
Introduction

- Search example (greedy algorithm)



Introduction

- Search example (greedy algorithm)



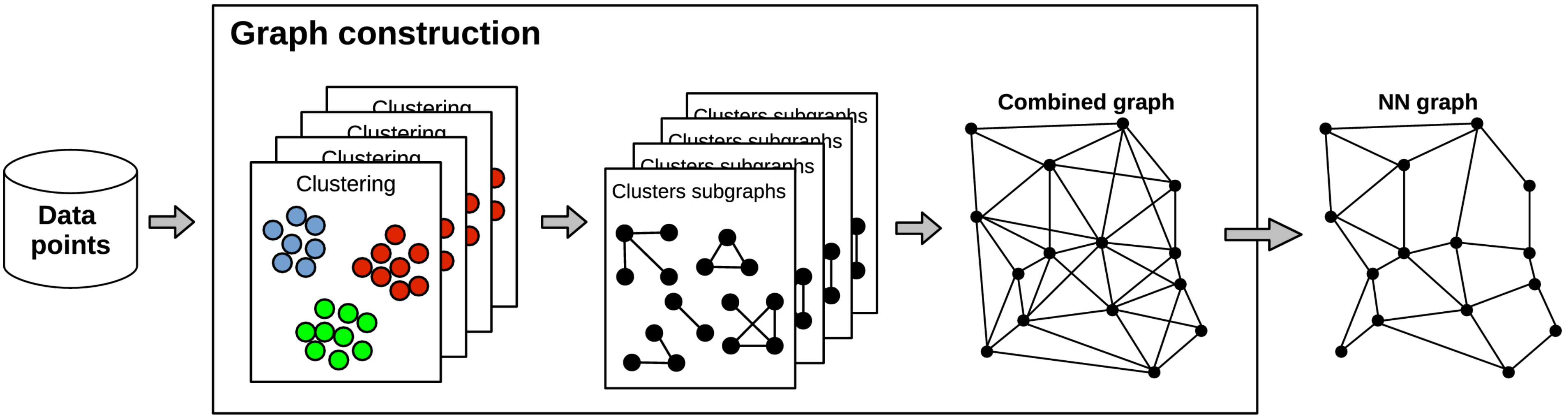
Hypothesis

H1. It is possible to construct a fast navigable NN graph based on multiple randomized clustering results

H2. It is possible to avoid exhaustive exploration of neighbors at some steps of search based on heuristics and learning techniques

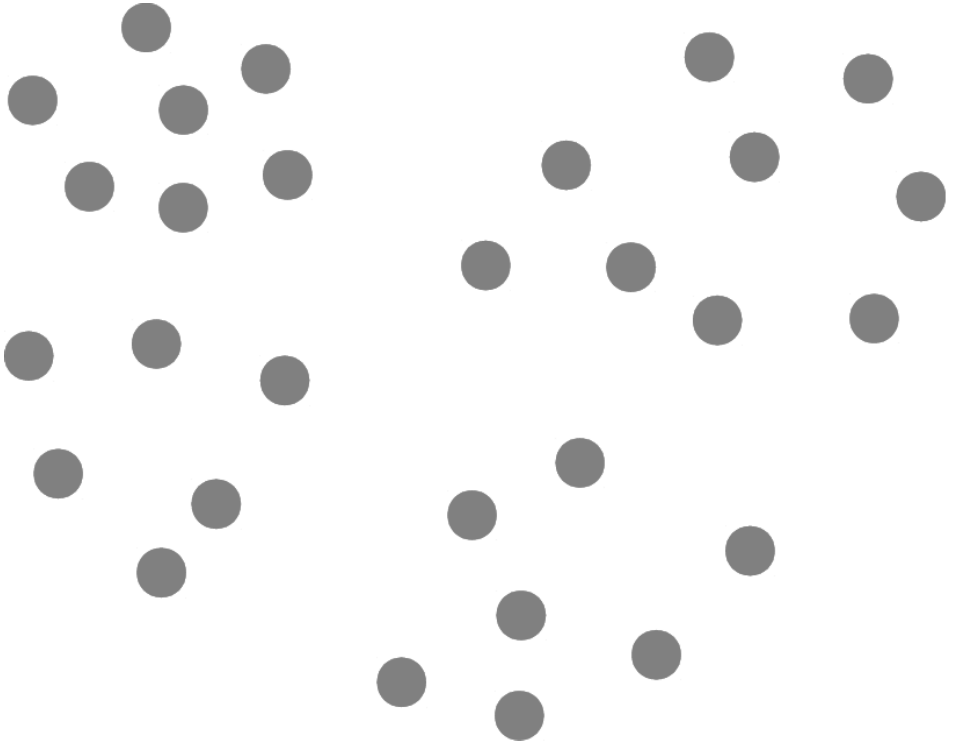
Methodology

- Framework: Hierarchical Clustering-based Nearest Neighbor Graph (HCNNG)



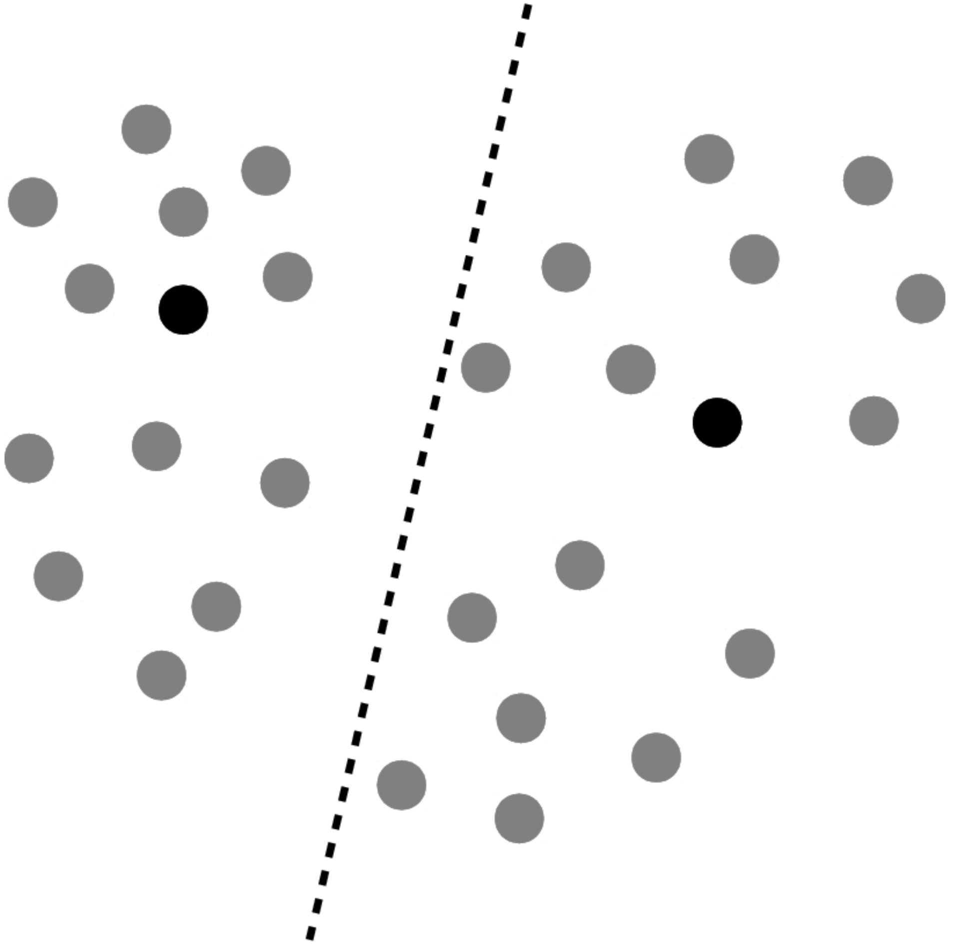
Methodology

- Hierarchical clustering



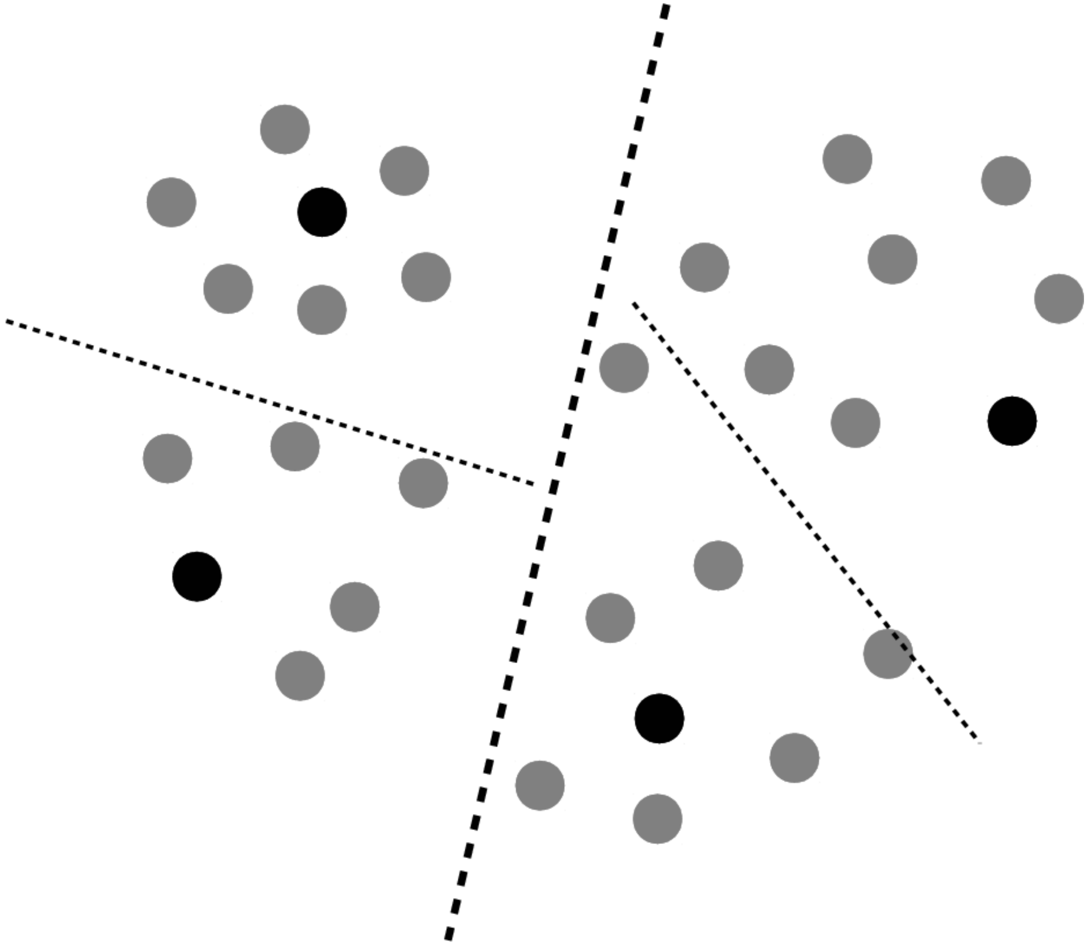
Methodology

- Hierarchical clustering



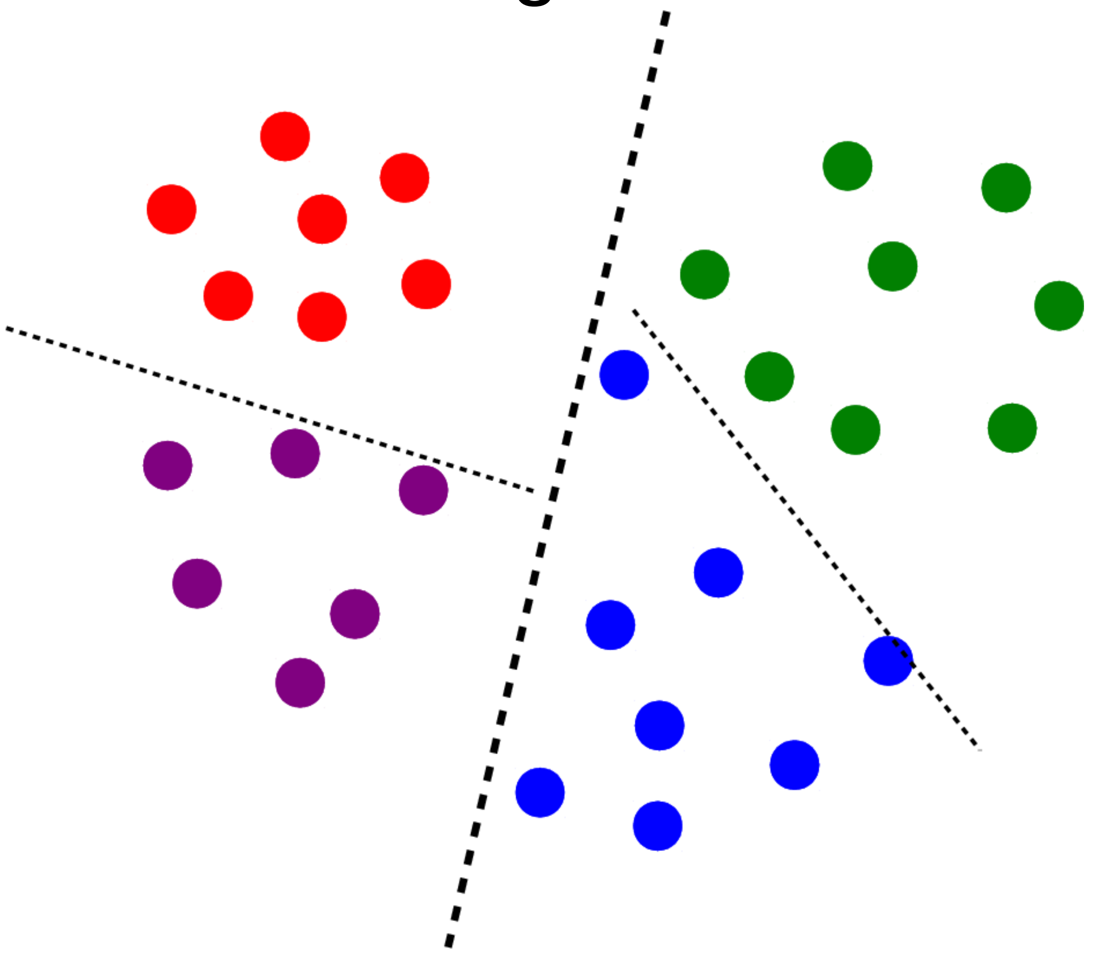
Methodology

- Hierarchical clustering



Methodology

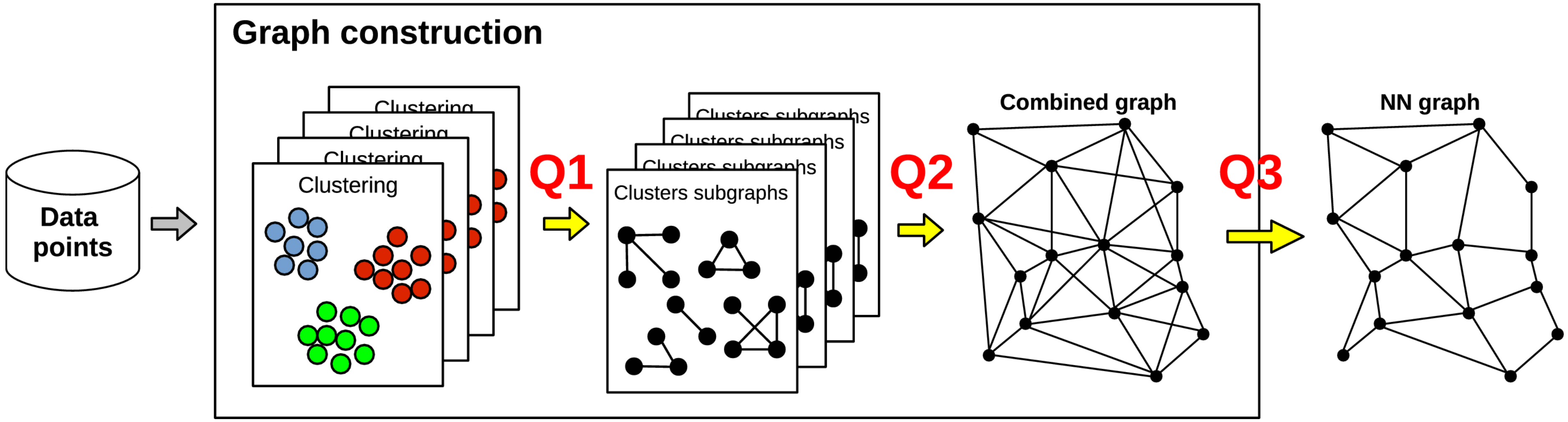
- Hierarchical clustering



Research Questions

Graph Construction:

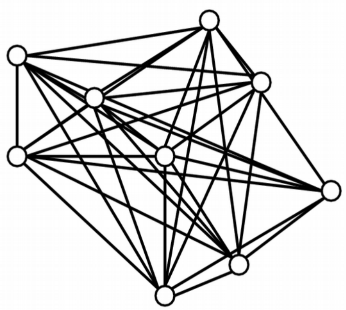
- **Q1.** What is the best way to connect points inside clusters?
- **Q2.** How can we merge subgraphs created in clusters?
- **Q3.** How could be improved the navigability in the NN graph?



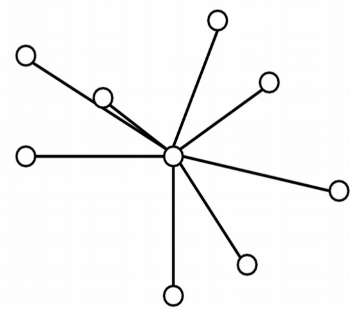
Research Questions

Q1. What is the best way to connect points inside clusters?

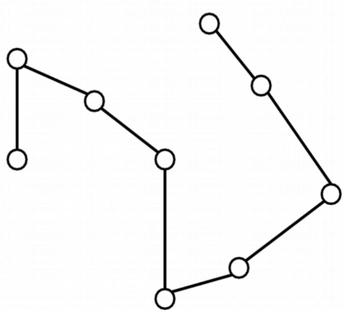
Complete graph



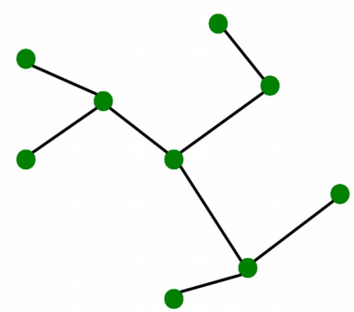
Star



Hamiltonian path



MST3



	Max degree	Max path
Complete	O(n)	O(1)
Star	O(n)	O(1)
Hamiltonian	O(1)	O(n)
MST3	**O(1)**	**log(n)**

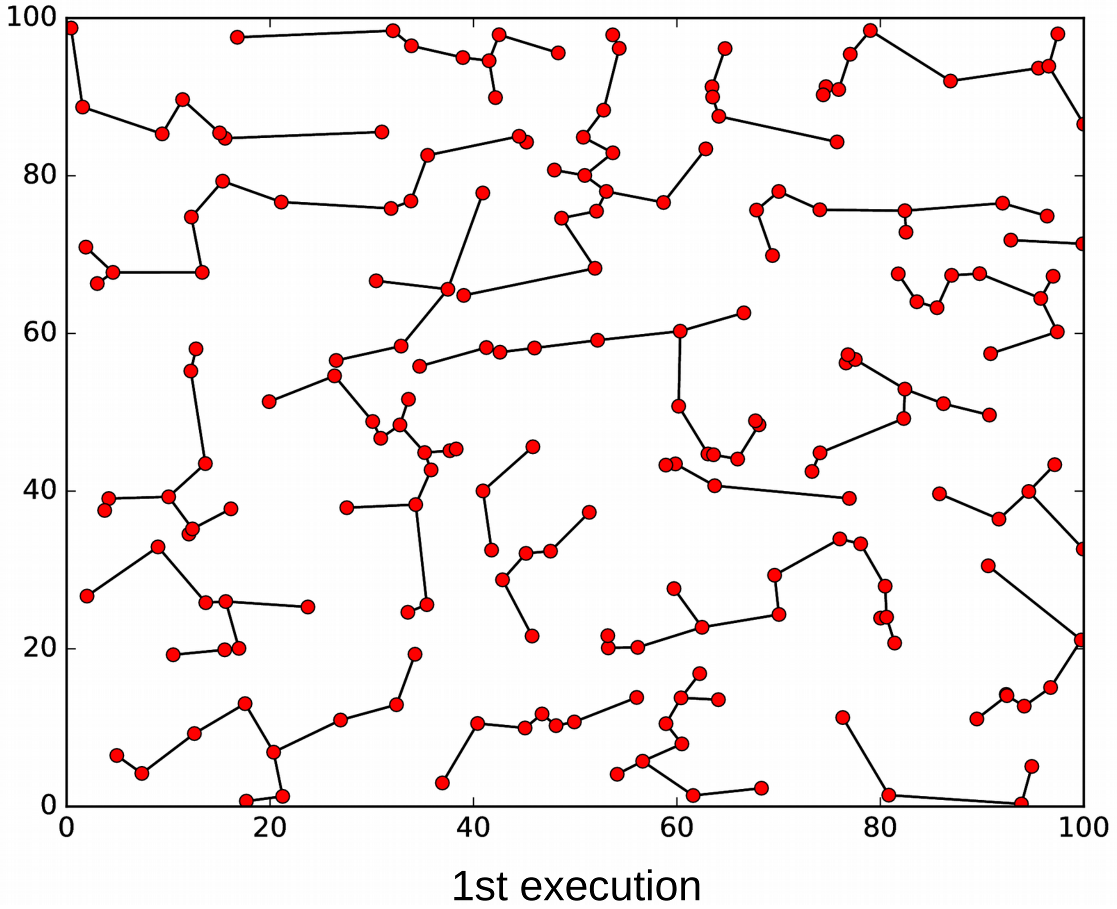
Research Questions

Q2. How can we merge subgraphs created in clusters?

– Make the union of vertices and edges of all subgraphs without repetition

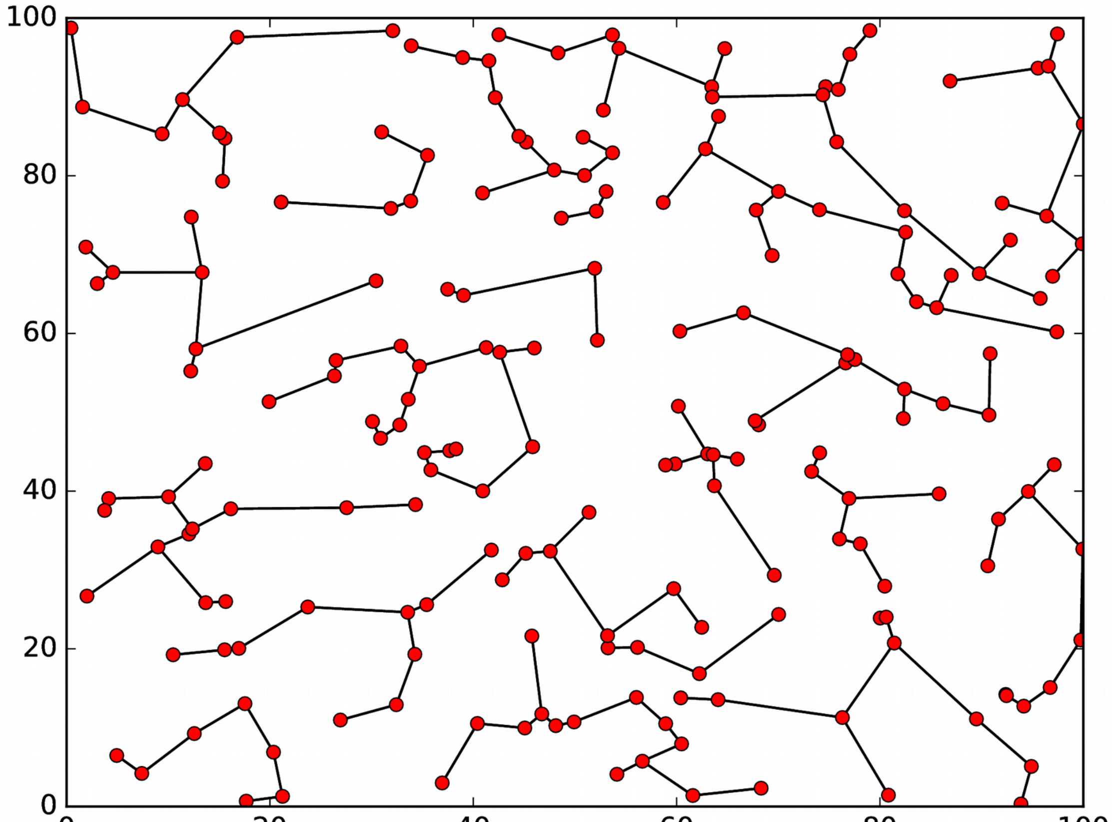
Research Questions

Q2. How can we merge subgraphs created in clusters?



Research Questions

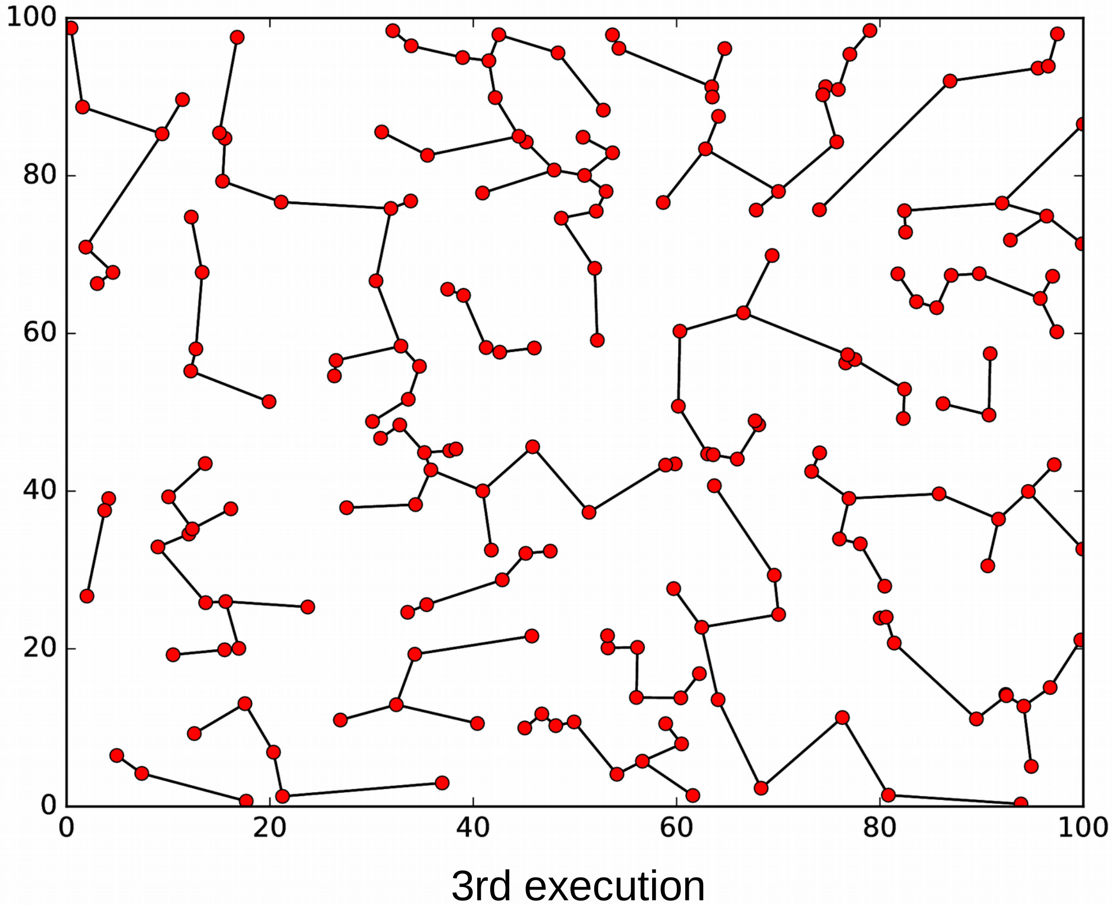
Q2. How can we merge subgraphs created in clusters?



2nd execution

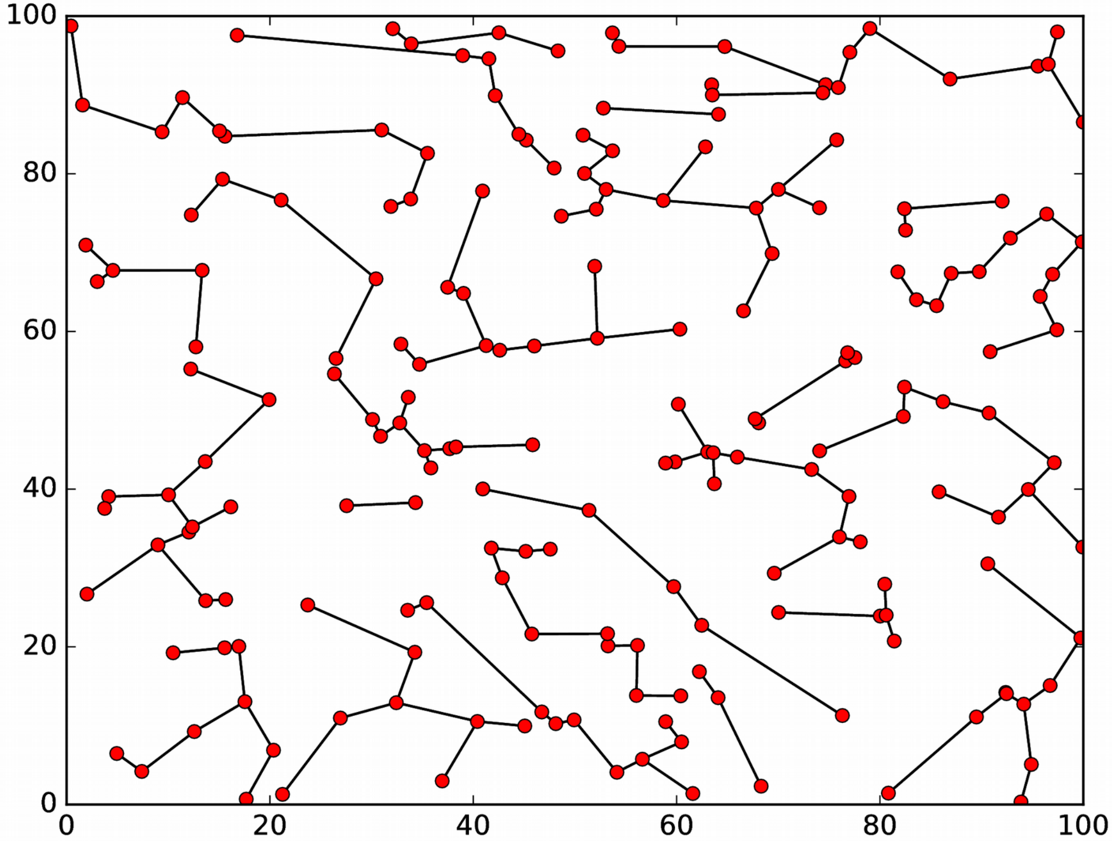
Research Questions

Q2. How can we merge subgraphs created in clusters?



Research Questions

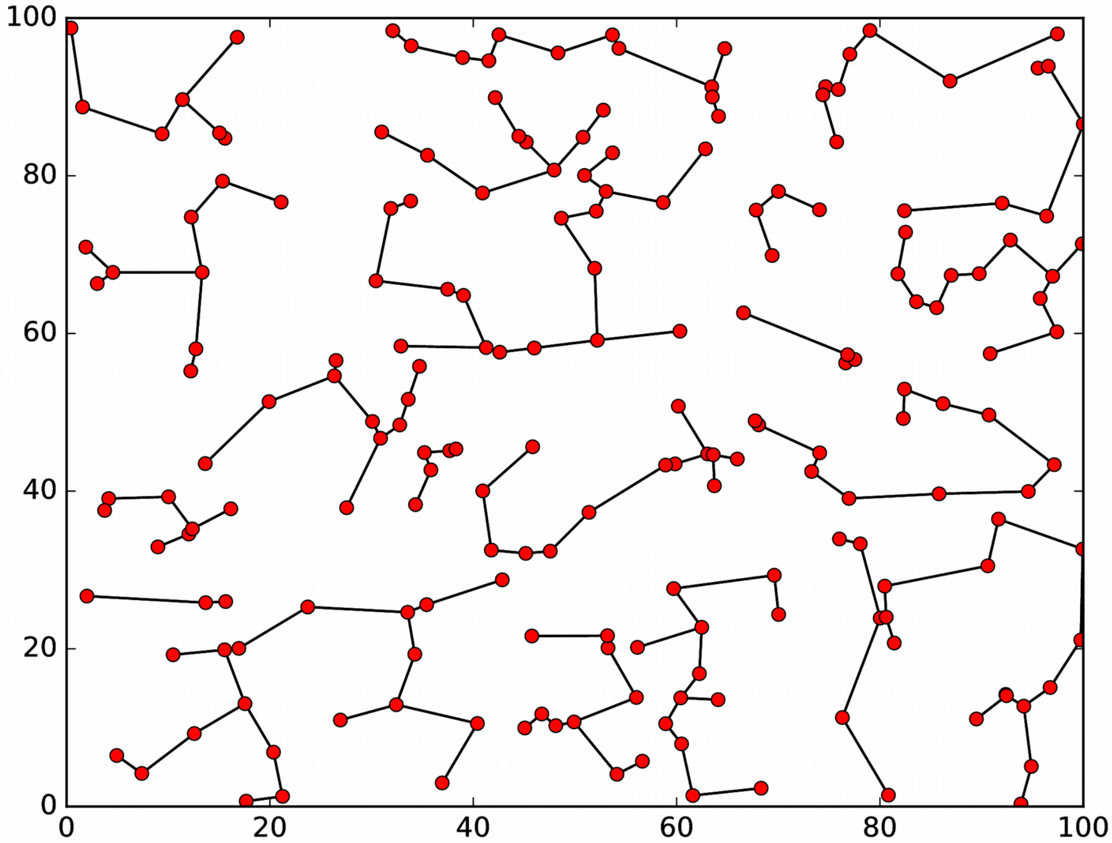
Q2. How can we merge subgraphs created in clusters?



4th execution

Research Questions

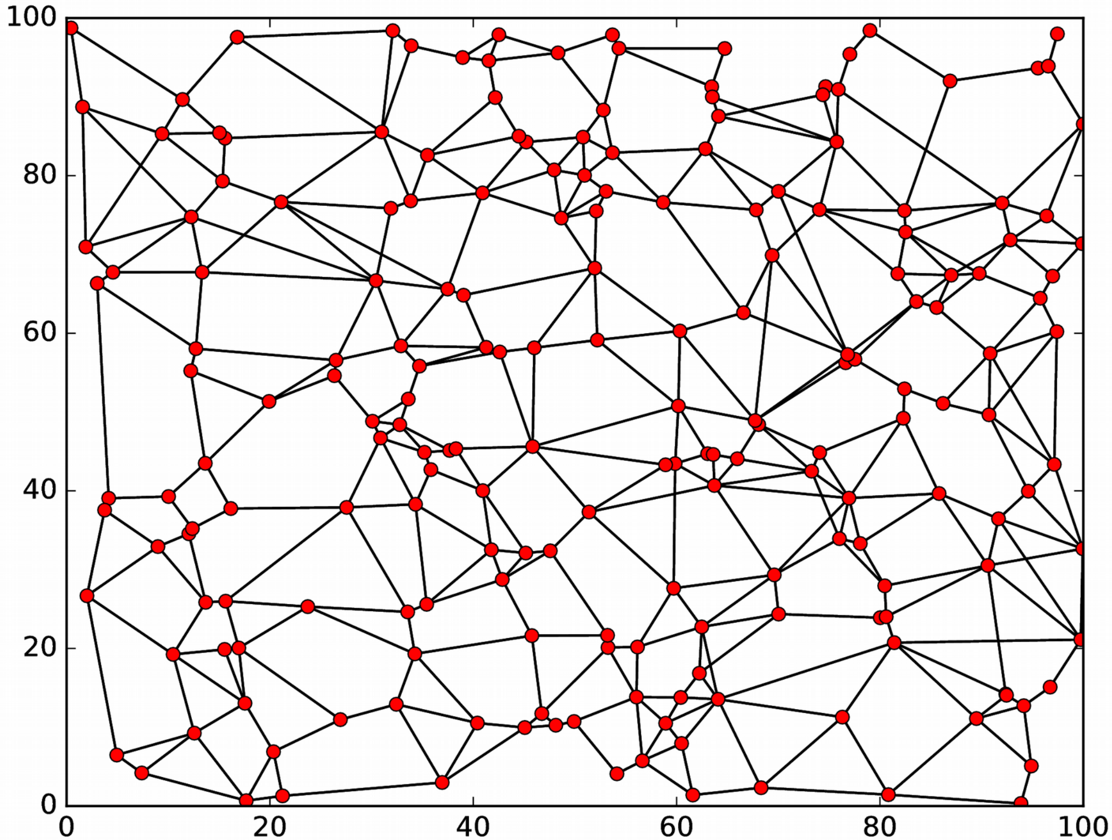
Q2. How can we merge subgraphs created in clusters?



5th execution

Research Questions

Q2. How can we merge subgraphs created in clusters?



Combined graph – after 15 executions

Research Questions

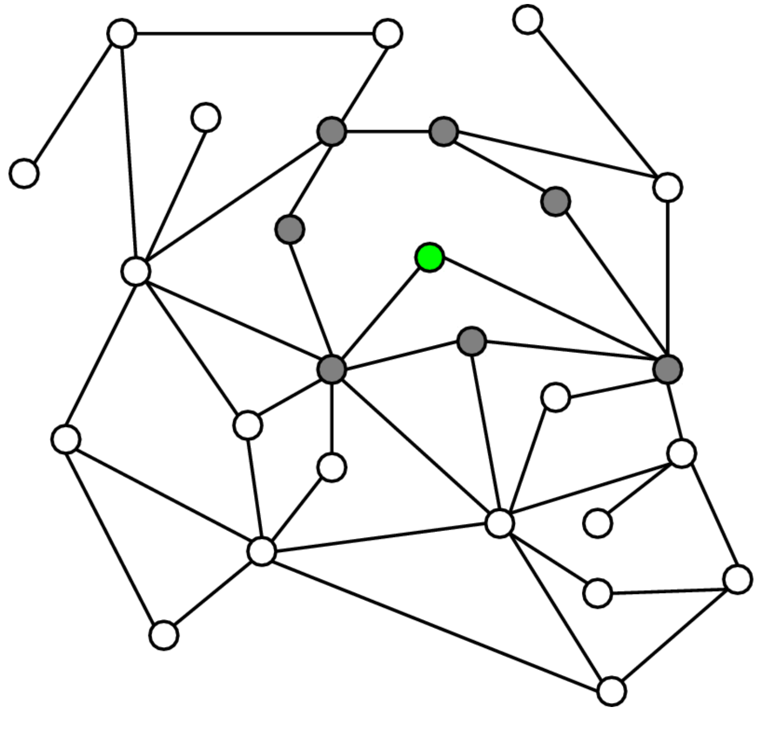
Q3. How could be improved the navigability in the NN graph?

– Under investigation

Research Questions

Q3. How could be improved even more the navigability in the NN graph?

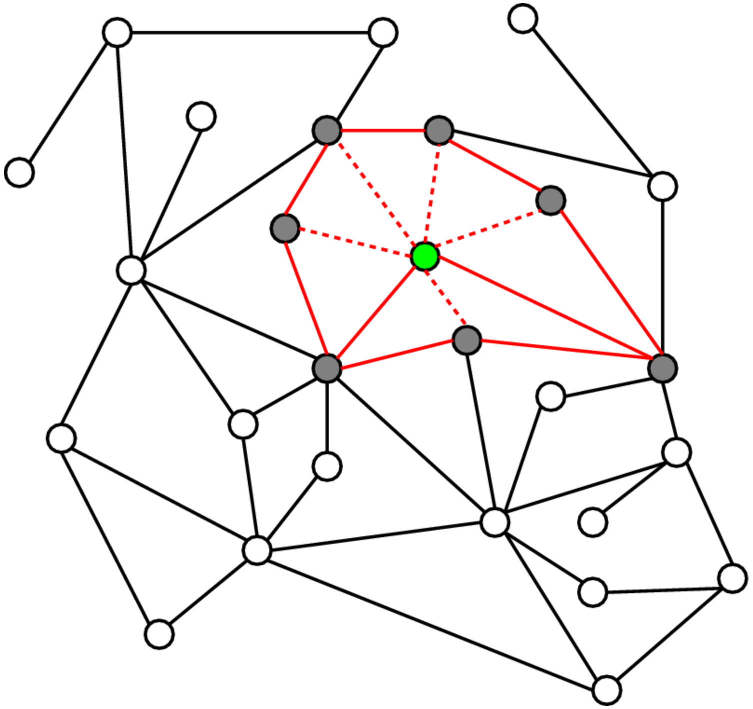
- Harwood and Drummond (2016)



Research Questions

Q3. How could be improved even more the navigability in the NN graph?

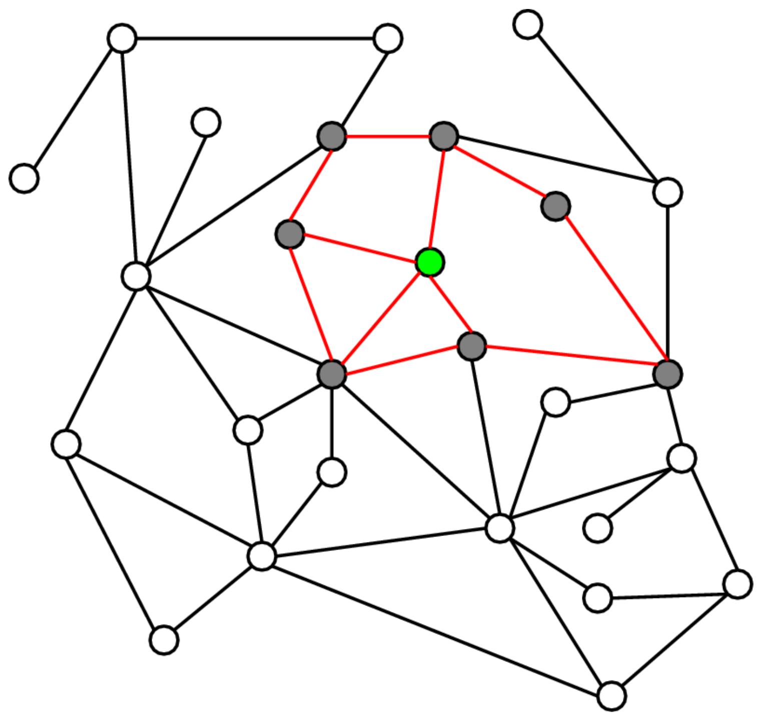
- Harwood and Drummond (2016)



Research Questions

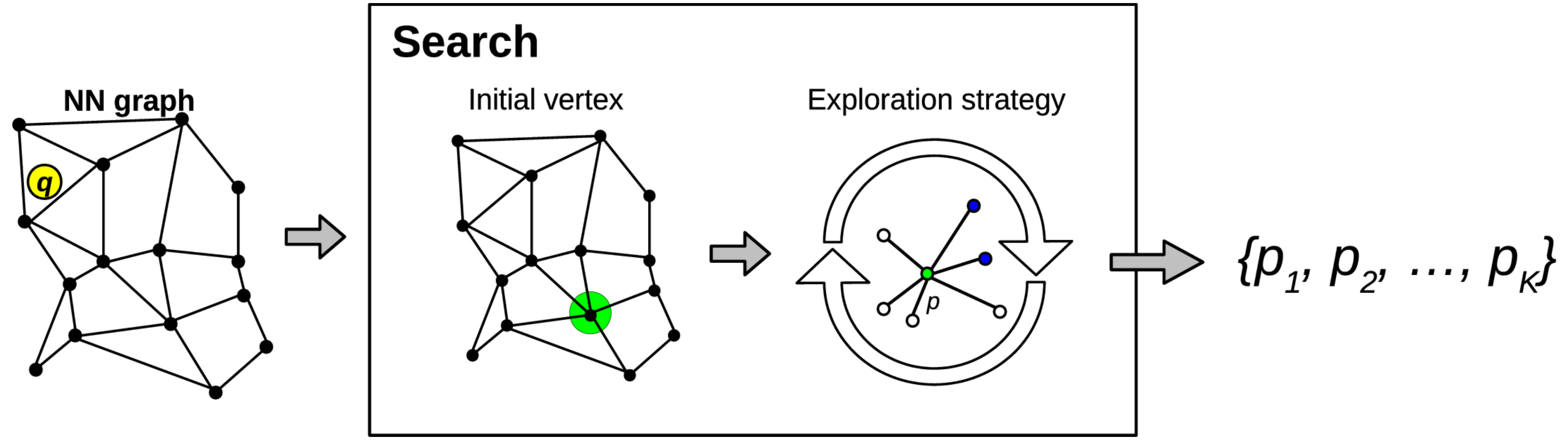
Q3. How could be improved even more the navigability in the NN graph?

– Harwood and Drummond (2016)



Methodology

- Framework: Hierarchical Clustering-based Nearest Neighbor Graph (HCNNG)



Research Questions

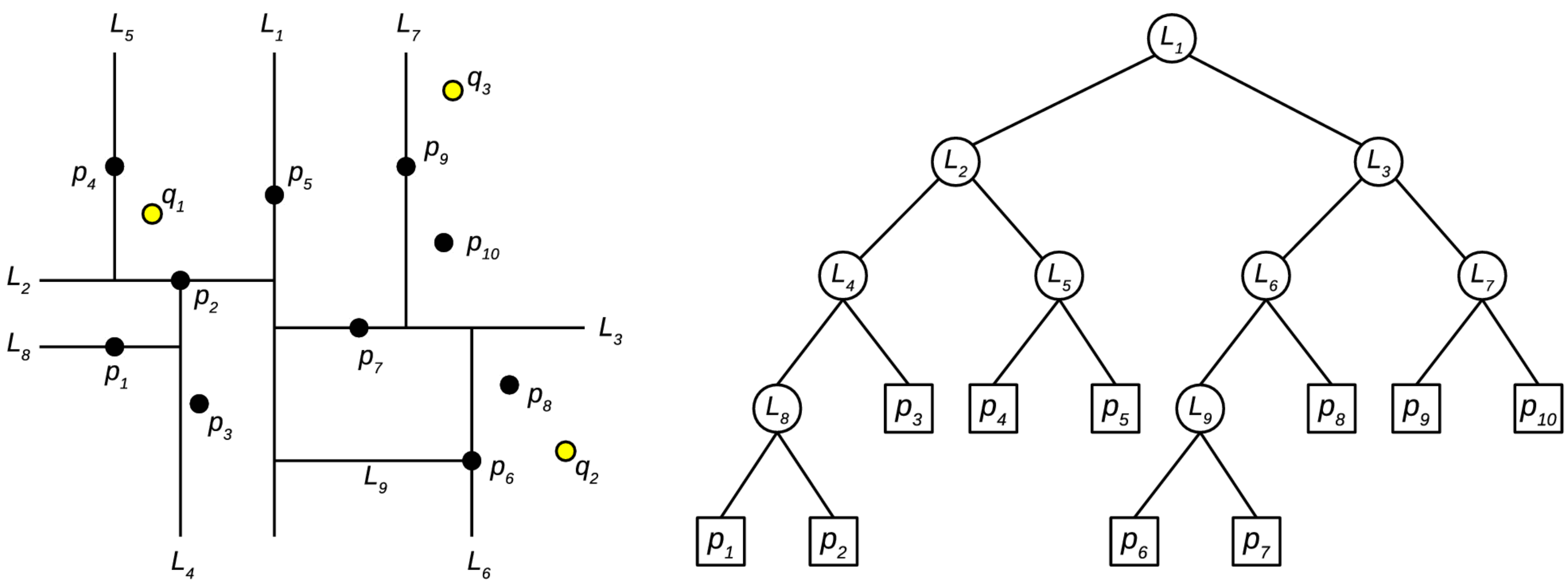
Search on NN graph:

- **Q4.** How to estimate a good starting vertex in a search?
- **Q5.** Which heuristic could be used to avoid exahustive exploration of neighbors at some steps of search?
- **Q6.** How could be applied a learning technique to avoid exahustive exploration of neighbors at some steps of search?

Research Questions

Q4. How to estimate a good starting vertex for search?

- Using leaf of a KD-Tree that contains the query point:



Examples:

$q_1 \longrightarrow p_5$

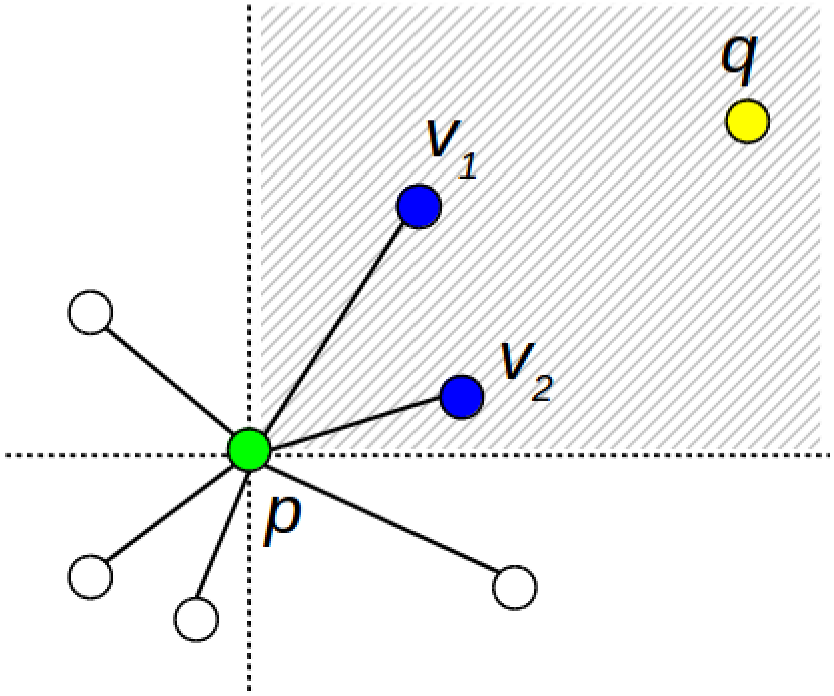
$q_2 \longrightarrow p_8$

$q_3 \longrightarrow p_{10}$

Research Questions

Q5. Which heuristic could be used to avoid exahustive exploration of neighbors at some steps of search?

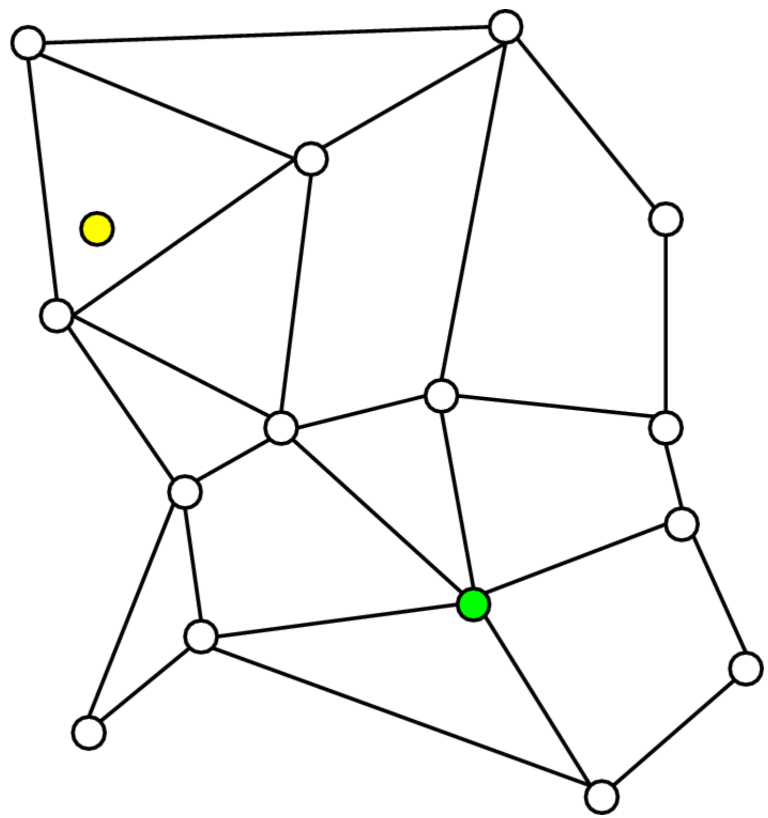
– Using definition of quadrants, ***guided search***:



Research Questions

Q5. Which heuristic could be used to avoid exahustive exploration of neighbors at early steps of search?

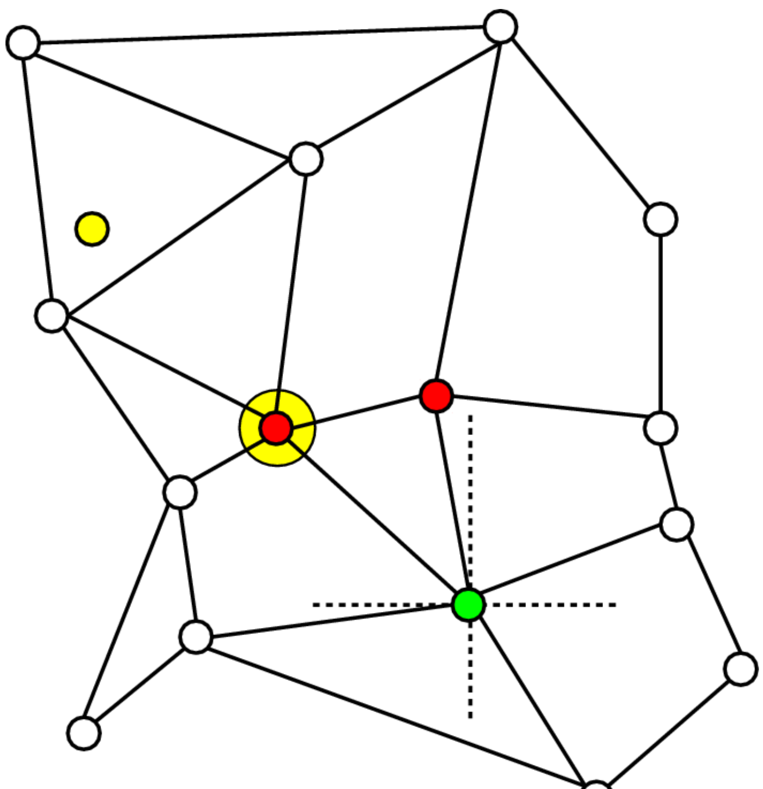
- Example of guided search:



Research Questions

Q5. Which heuristic could be used to avoid exahustive exploration of neighbors at early steps of search?

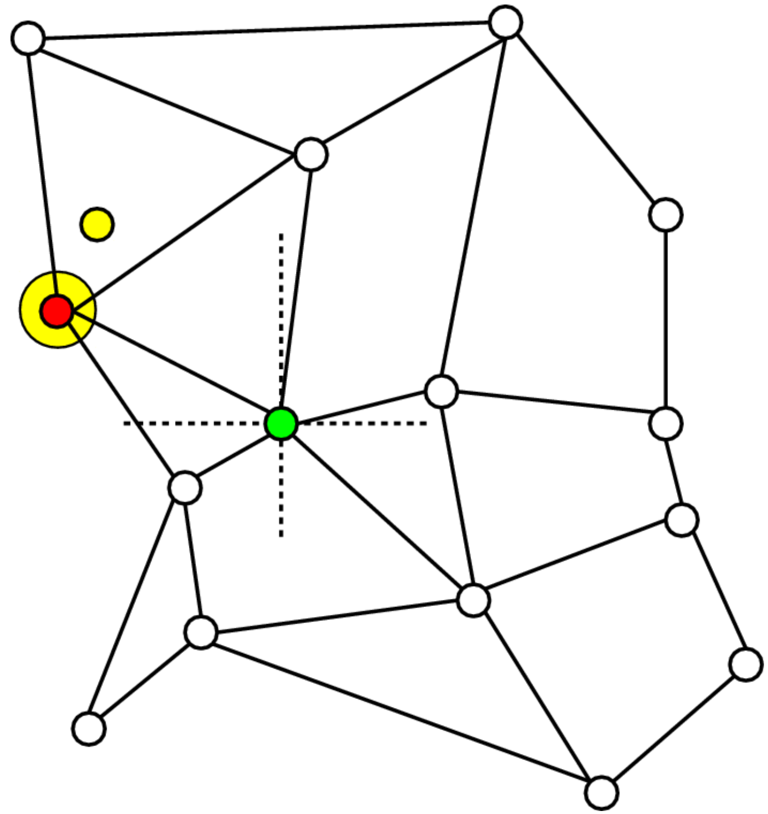
- Example of guided search:



Research Questions

Q5. Which heuristic could be used to avoid exahustive exploration of neighbors at early steps of search?

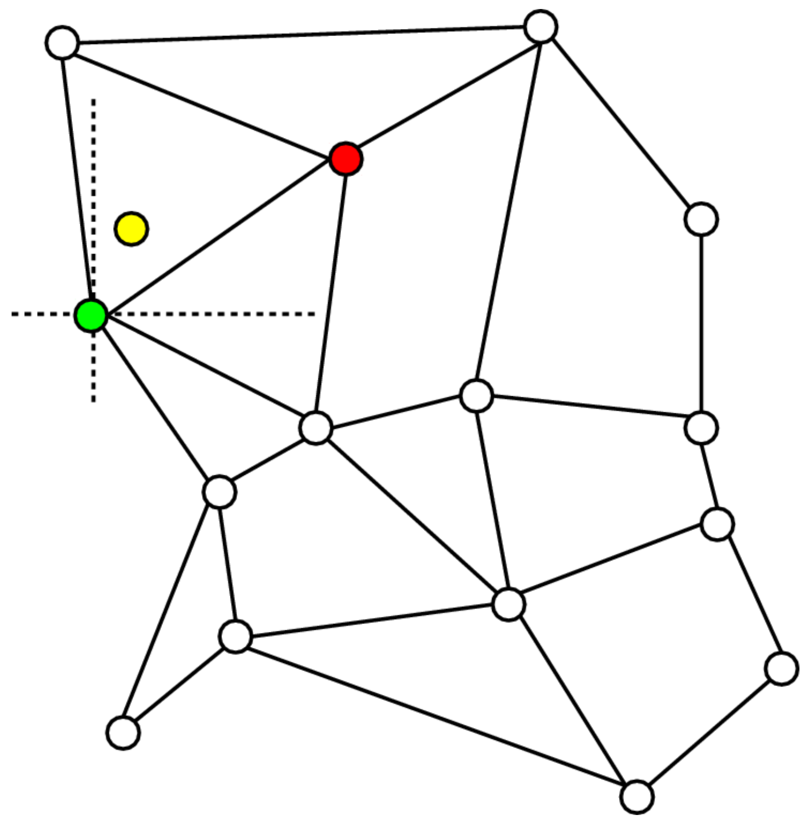
– Example of guided search:



Research Questions

Q5. Which heuristic could be used to avoid exahustive exploration of neighbors at early steps of search?

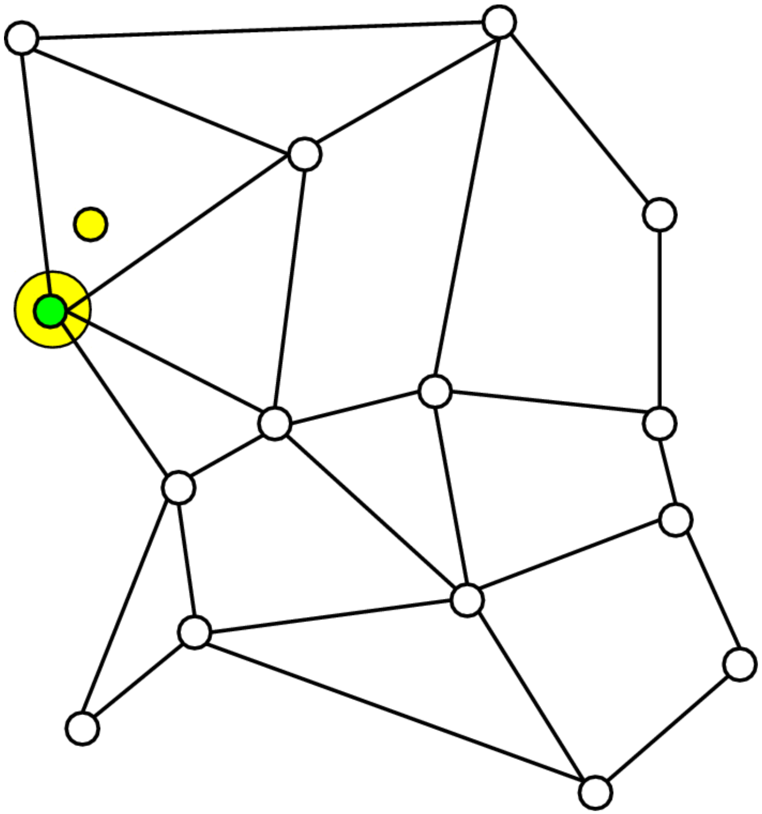
– Example of guided search:



Research Questions

Q5. Which heuristic could be used to avoid exahustive exploration of neighbors at early steps of search?

– Example of guided search:

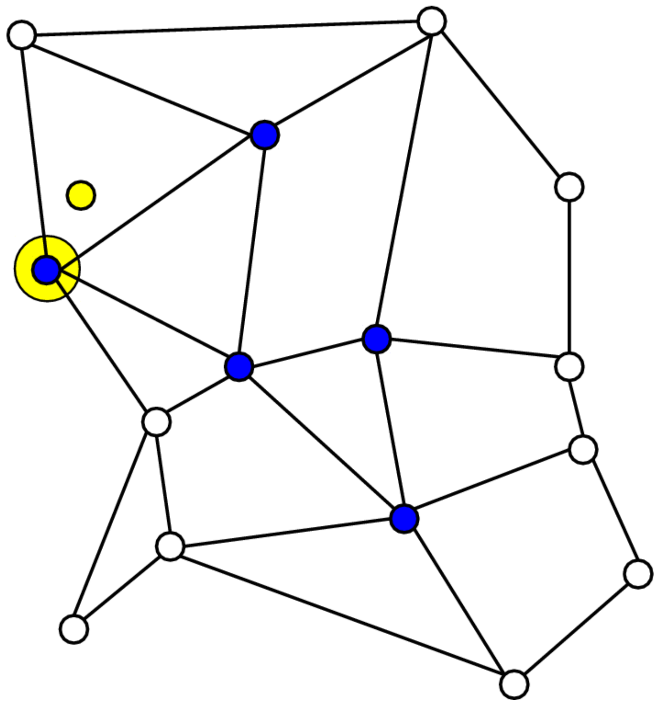


Research Questions

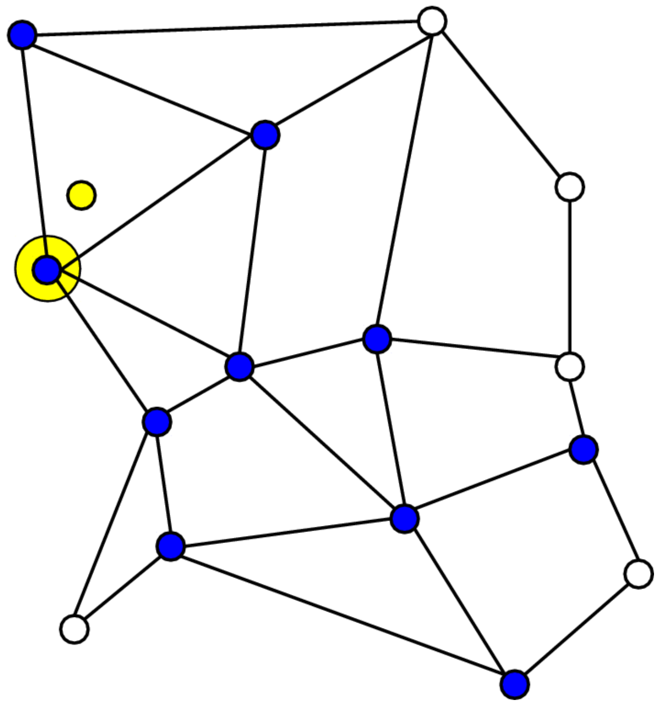
Q5. Which heuristic could be used to avoid exahustive exploration of neighbors at early steps of search?

– Example of guided search:

guided search



greedy search



Research Questions

Q6. How could be applied a learning technique to avoid exahustive exploration of neighbors at some steps of search?

- Under investigation

HCNNG vs State-of-the-art

	HCNNG	HNSW	FANNG	SW	KGraph
Graph construction					
# graphs	**single**	multiple	single	single	single
strategy	**divide and conquer**	incremental construction	incremental construction	incremental construction	optimization of random graph
generic space	**yes**	yes	no	yes	yes
Search					
initial vertex	**KD-Tree based**	random	centroid	multiple random	random
strategy	**guided + backtracking**	multiple level + backtracking	backtracking	backtracking	backtracking

Validation

- BIGANN datasets for ANNS (visual features):
  - 1 million of SIFT features vectors (128 dimensions) to index construction, and 10K queries to evaluate performance
  - 1 million of GIST features vectors (960 dimensions) to index construction and 1K queries to evaluate performance
  - These datasets were previously used in other recent works to evaluate ANNS techniques
- GloVe (textual features)
  - 1 million of GloVe features vectors (100 dimensions) to index construction, and 10K queries to evaluate performance

Validation

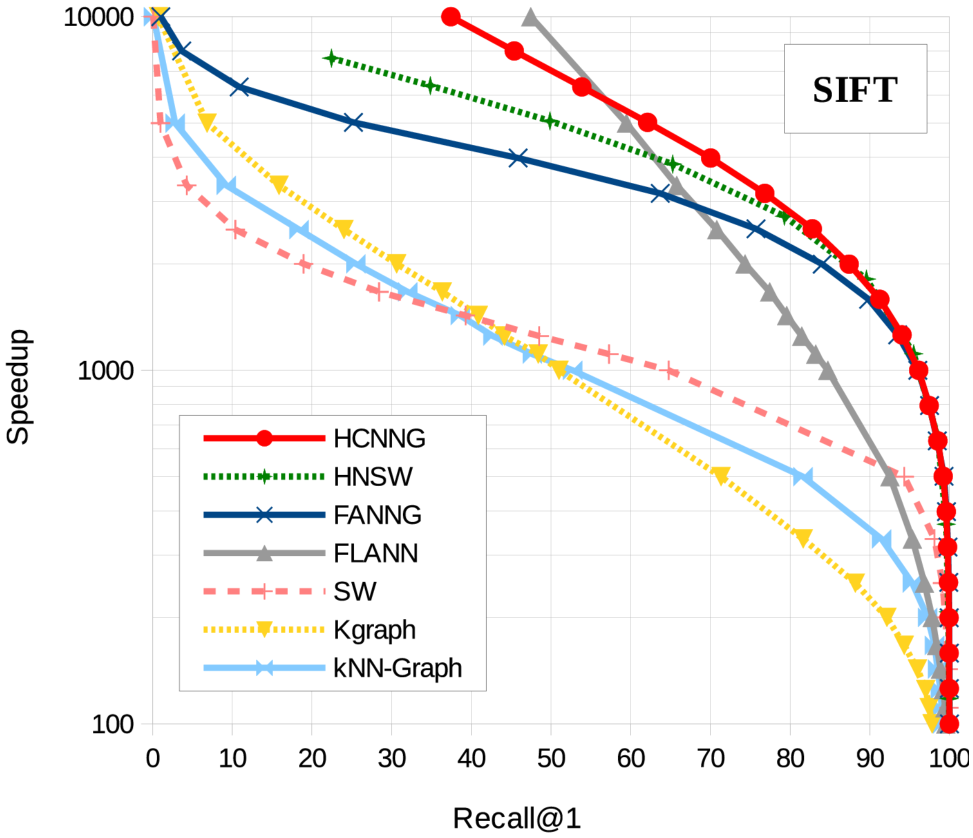
- Baselines
  - Fast Library for Approximate Nearest Neighbors (**FLANN**, Muja and Lowe, 2014), a well-known and widely used
    - Randomized KD-Trees, K-Means Tree, and Hierarchical Clustering Tree
    - Auto-tuned algorithm, which selects the best algorithm and parameter values based on the data
  - Fast Approximate Nearest Neighbour Graphs (**FANNG**, Harwood and Drummond, 2016): Using our own implementation
  - Small world graphs (**SW**, Malkov et al., 2013) and Hierarchical Navigable Small World (**HNSW**, Malkov and Yashunin, 2017): Using the implementation found in Non-Metric Space Library (NMSLIB)
  - **Kgraph** (Dong et al., 2011): Using the implementation provided by authors

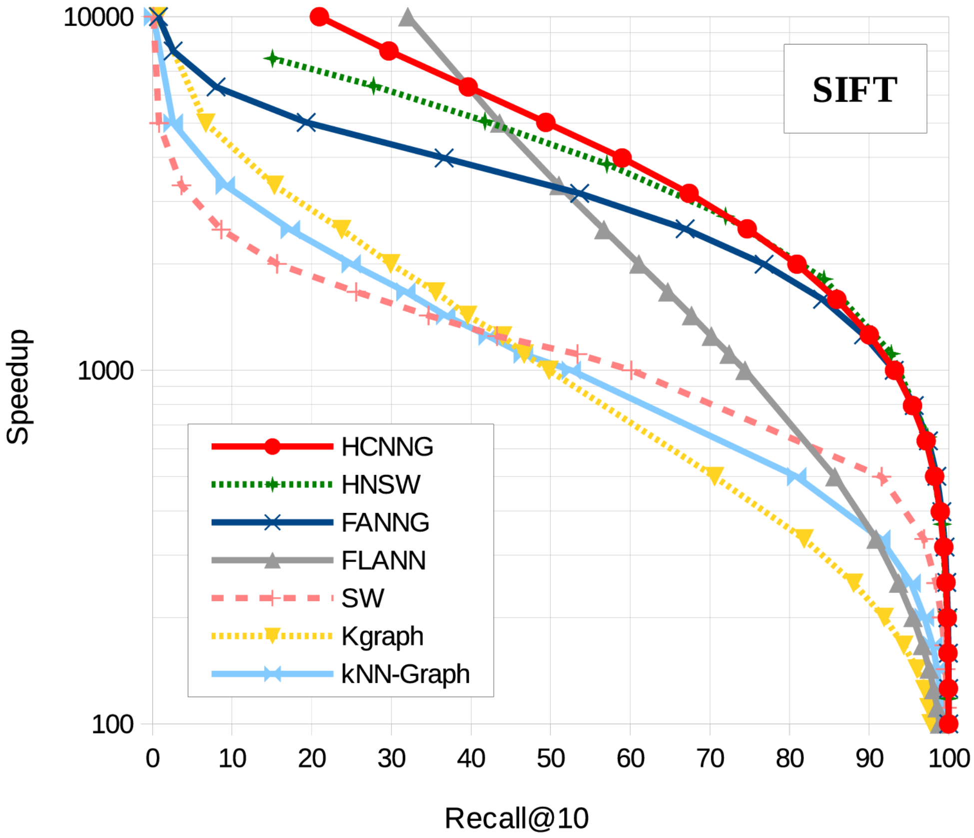
Validation

- We employed a widely-used evaluation measure for ANNS like the *Speedup x Recall charts*
- To keep the speedup independent from architecture where experiments are executed, we will only consider the number of distance calculations performed by each method. Thus, speedup is defined by:

$$Speedup = \frac{Collection\ size}{Number\ of\ distance\ calculations}$$

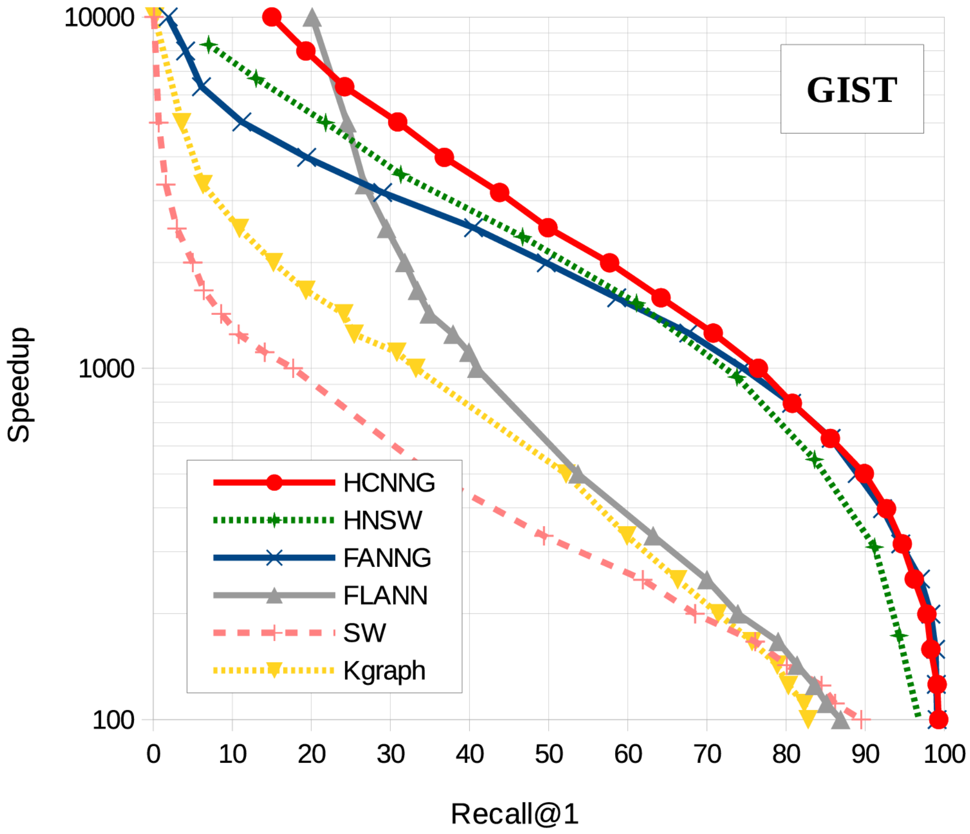
Preliminary Results

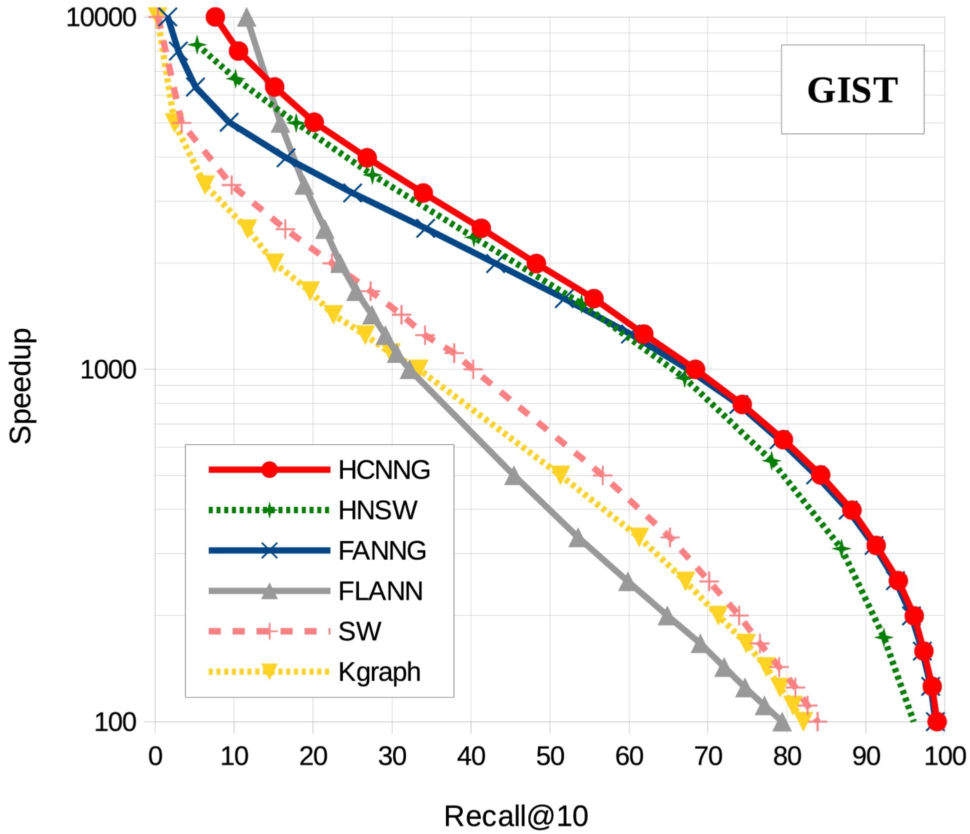




Results for 1-NN and 10-NN search on 1M SIFT dataset

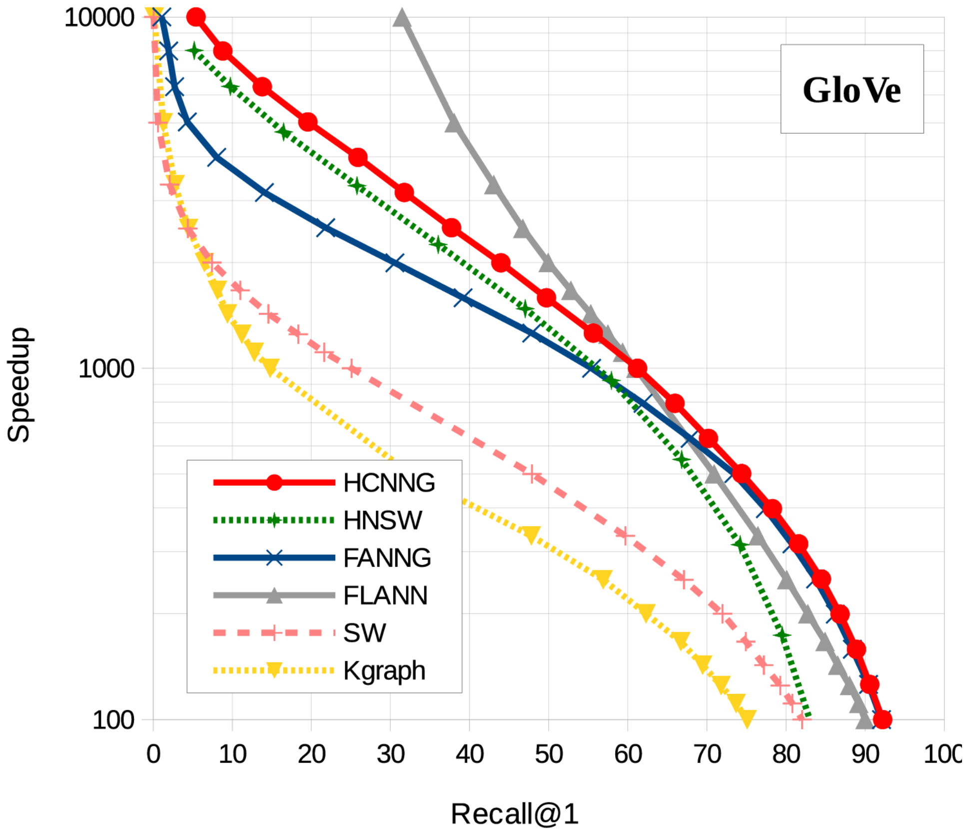
Preliminary Results

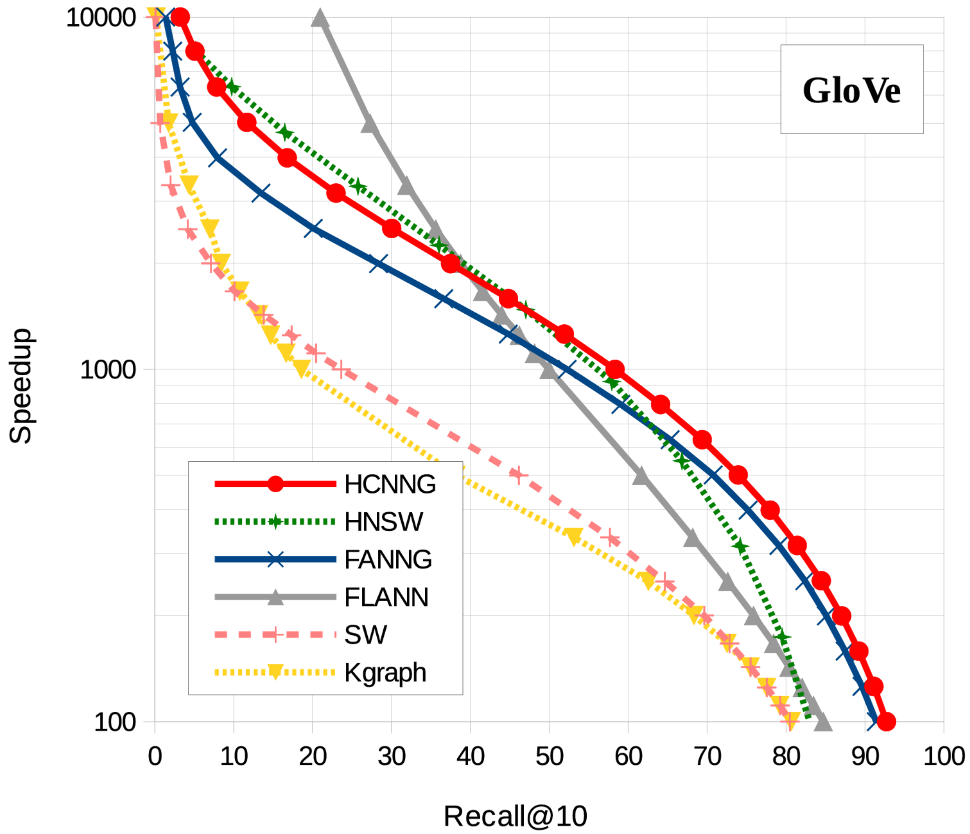




Results for 1-NN and 10-NN search on 1M GIST dataset

Preliminary Results





Results for 1-NN and 10-NN search on 1M Glove dataset

Schedule

1. Courses and initial literature review.
2. NN graph construction algorithm.
3. Techniques for removing edges redundancy and reinforce connectivity on NN graphs.
4. Heuristics for guided search on NN Graphs.
5. Learning techniques for guided search on NN Graphs.
6. Internship (PhD sandwich program with BEPE scholarship).
7. Validation of framework and publication of the main results.
8. Writing and defense of the PhD work.

Activity	Semester							
	1s2016	2s2016	1s2017	2s2017	1s2018	2s2018	1s2019	2s2019
1	•	•						
2		•	•					
3			•	•	•			
4			•	•	•			
5					•	•	•	
6					•	•		
7					•	•	•	•
8								•

References

Andoni, A., & Indyk, P. (2006). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on (pp. 459-468). IEEE.

Dong, W., Moses, C., & Li, K. (2011). Efficient k-nearest neighbor graph construction for generic similarity measures. In Proceedings of the 20th international conference on World wide web (pp. 577-586). ACM.

Harwood, B., & Drummond, T. (2016). FANNG: fast approximate nearest neighbour graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5713-5722).

Lv, Q., Josephson, W., Wang, Z., Charikar, M., & Li, K. (2007, September). Multi-probe LSH: efficient indexing for high-dimensional similarity search. In Proceedings of the 33rd international conference on Very large data bases (pp. 950-961). VLDB Endowment.

Malkov, Y., Ponomarenko, A., Logvinov, A., & Krylov, V. (2014). Approximate nearest neighbor algorithm based on navigable small world graphs. Information Systems, 45, 61-68.

Malkov, Y., & Yashunin, D. A. (2016). Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. arXiv preprint arXiv:1603.09320.

Muja, M., & Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(11), 2227-2240.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Thanks!