A Simulation Study of Walks in Large Social Graphs

by

Shahed Anwar
B.Sc., University of Dhaka, 1999
M.Sc., University of Dhaka, 2001

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

A Simulation Study of Walks in Large Social Graphs

by

Shahed Anwar
B.Sc., University of Dhaka, 1999
M.Sc., University of Dhaka, 2001

Supervisory Committee

Dr. Kui Wu, Co-Supervisor
(Department of Computer Science, University of Victoria)

Dr. Jianping Pan, Co-Supervisor
(Department of Computer Science, University of Victoria)

Supervisory Committee

Dr. Kui Wu, Co-Supervisor
(Department of Computer Science, University of Victoria)

Dr. Jianping Pan, Co-Supervisor
(Department of Computer Science, University of Victoria)

ABSTRACT

Online Social Networks (OSNs) such as Facebook, Twitter, and YouTube are among the most popular sites on the Internet. Billions of users are connected through these sites, building strong and effective communities to share views and ideas, and make recommendations nowadays. Therefore, by choosing an appropriate user-base from billions of people is required to analyze the structure and key characteristics of the large social graphs to improve current systems and to design new applications. For this reason, node sampling technique plays an important role to study large-scale social networks. As a basic requirement, the sampled nodes and their links should possess similar statistical features of the original network, otherwise the conclusion drawn from the sampled network may not be appropriate for the entire population. Hence, good sampling strategies are key to many online social network applications. For instance, before introducing a new product or adding new feature(s) of a product to the online social network community, that specific new product or the additional feature has to be exposed to only a small set of users, who are carefully chosen to represent the complete set of users. As such, different random walk-based sampling techniques have been introduced to produce samples of nodes that not only are internally well-connected but also capture the statistical features of the whole network. Traditionally, walk-based techniques do not have the restriction on the number of times that a node can be re-visited while sampling. This may lead to an inefficient sampling method, because the walk may be "stuck" at a small number of high-degree nodes without being able to reach out to the rest of the nodes. A random walk, even

after a large number of hops, may not be able to obtain a sampled network that captures the statistical features of the entire network.

In this thesis, we propose two walk-based sampling techniques to address the above problem, called K-Avoiding Random Walk (KARW) and Neighborhood-Avoiding Random Walk (NARW). With KARW, the number of times that a node can be re-visited is constrained within a given number K. With NARW, the random walk works in a "jump" fashion, since the walk starts outside of the N-hop neighborhood from the current node chosen randomly. By avoiding the current nodes neighboring area of level-N, NARW is expected to reach out the other nodes within the entire network quickly. We apply these techniques to construct multiple independent subgraphs from a social graph, consisting of 63K users with around a million connections between users collected from a Facebook dataset. By simulating our proposed strategies, we collect performance metrics and compare the results with the current state-of-the-art sampling techniques (Uniform Random Sampling, Random Walk, and Metropolis Hastings Random Walk). We also calculate some of the key statistical features (i.e., degree distribution, betweenness centrality, closeness centrality, modularity, and clustering coefficient) of the sampled graphs to get an idea about the network structures that essentially represent the original social graph.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

Dr. Kui Wu and Dr. Jianping Pan for their constant support in my academic and research areas, providing me valuable feedback about my research topic, which helped me improve my inner potential to accomplish my goals. I would also like to thank them for allowing and recommending me to go for co-op for a year to work in the Desktop Transformation Project (DTP) team, the second largest project at Provincial Health Services Authority (PHSA) in BC. It gave me great opportunities not only to have an experience in the Canadian professional environment but also to open up doors for my future career. I am so thankful to Dr. Kui Wu and Dr. Jianping Pan to believe in me that I can finish my research and can successfully graduate this year and move forward.

My Wife for giving me inspiration and support both mentally and financially throughout this journey. Besides being an architect, she stepped up and contributed significantly during our financial hardship, starting a full-time job in a bakery, working 4/5 days a week from 4:00 am to make ends meet until I headed for coop in Sept 2013. She also took great care of us in preparing foods and household works everyday. She is a great source of inspiration and a huge mental support for me and for her relentless support, I was able to continue my academic career without getting stressed out and now I'm so close to my graduation.

University of Victoria for giving me the opportunity to study MSc. with a UVic fellowship from Sept, 2011 – Aug, 2012. Without this support, I would not have been in Canada studying Computer Science. I am grateful to the fellowship committee for this great opportunity.

Last but not least, I would like to thank my loving parents and in-laws who are always being there for me and always keep us in their prayers. Without their blessings, I would not have gone this far. I am immensely grateful to my beloved friends and their families in Victoria and Vancouver for their encouragement during these years.

sincerely, Shahed Anwar

DEDICATION

Dedicated to my beloved wife, Nafisa Shahrin Bari and only son, Zawata Afnaan.

Chapter 1

Introduction

1.1 Social Networks

Online Social Networks (OSNs) have become immensely popular nowadays and the study of social graphs has attracted a large number of researchers all over the world. According to statistics [3] as of January 2015, it has been estimated that about 42% of active population around the world use the Internet and among the active internet users, more than 65% use OSNs actively. It has been projected that the user of OSNs would grow upto around 2.44 billion by the end of 2018 [4]. One of the biggest challenges of studying social graphs is its mammoth size and dynamic structure that changes over time. Due to the massive size and dynamic nature of the complex and scale-free graphs, it becomes almost impossible to analyze the structure of the entire social graph. Therefore, graph sampling techniques have been evolved over the years to obtain smaller subgraphs that can essentially inherit almost all the characteristics of the original graph.

OSNs have been popular all over the world noticeably in recent years because of the fast spreading of messages and information over OSNs. Social media such as Facebook, Twitter, YouTube, WikiLeaks, Wikipedia, QQ (Chinese Social Media), Tuenti (Spanish Social Media), and Naver (Korean Social Network) have been used extensively for social interactions, news, marketing products, rumors and political purposes. According to Facebook statistics [1], there are 936 million daily active users on average and in Twitter [2], it has been estimated that 302 million active users around the world generate 500 million "tweets" a day. To understand the information propagation over OSN [26] and the social interaction [44], and to characterize user-

behaviors [10], we need to investigate the topological features of social networks. For this purpose, sampling algorithms are designed to obtain a smaller social graph that exhibits same or similar topological features as the original graph.

1.1.1 Motivation

Billions of people around the world are using social networking sites and online social networking has become a part of their lives. With OSNs, it is the most effective way to interact with people of similar or different communities to share views and ideas, not only within communities but also across the boundary to reach out people around the world. In addition, the number of people that can be interacted with through social media has a huge benefit and significant impact for promoting business product(s) nowadays, because the cost involved using social media and social networking is considerably lower than that of the traditional marketing and advertising strategies. It has now become very much essential to select potential customers from online social networks to use new product(s) and expect unbiased feedback about features in order to decide whether to keep or enhance the quality before introducing to the entire community. Satisfying users among the consumer community can ultimately promote business by the referring the newly introduced product(s) to their friends and kins through online social networks. For this reason, companies around the world have adapted to use social media as their business promotion platform and follow this marketing strategy of introducing their product prototype to a small number of users before releasing it to the actual market. In this way, the companies can greatly reduce risk and save product features enhancement, modification cost before final release. Nevertheless, such practice is meaningful only when the selected small user base can effectively inherit all the statistical features to represent the large customer community. As such, good sampling method(s) is/are necessary for selecting a representative test-case user community.

Sampling technique plays a vital role for selecting potential users among millions in the social network. At times, selecting friends of certain users may end up getting similar feedback/criticism, which in turn has a good chance to become more biased as friends of users in social networks tend to have similar mentality and taste. Such biasness may be misleading when it comes to product development and marketing. For this reason, it is necessary to choose effective users who can potentially give unbiased feedback, help building a product with the highest standard over time.

From the sampling point of view, it may not be helpful if the sampled nodes are only concentrated in a small region even after so many times of sampling. In other words, an ideal and effective sampling technique would always sample users that are uniformly distributed over the network, resulting an ideal representation of the whole population.

1.2 Main Contributions

The main contributions of this thesis include two newly proposed sampling methods, called K-Avoiding Random Walk (KARW) and Neighborhood-Avoiding Random Walk (NARW). In our first method, KARW, we limit the number of re-visits to a node, at most $K-1$ times and avoiding the node K^{th} times during a walk. The intuition behind KARW is to avoid re-visiting nodes several times within a region and enforce the random walk to new regions. In the second method, we propose NARW, where N stands for the level-N of neighborhood nodes to be avoided during the walk. In this strategy, a list of neighborhood nodes upto level-N is to be generated before the walk begins and avoids them intentionally to "push" sampling nodes from different regions in the graph. When N is larger than 2, NARW actually works in a "jump" fashion, since the next sampled node is several hops away from the current node.

We compare our new methods with the three existing state-of-the-art algorithms, namely, Uniform Random Sampling (URS), Random Walk (RW), and Metropolis Hastings Random Walk (MHRW). After simulation, we compare results of KARW and NARW with the three most popular sampling techniques that are mentioned before, based on certain performance criteria, for instance, the actual length a walk is able to finish, number of total nodes sampled, number of nodes sampled uniquely while walking on the social graph. We also analyze statistical key features of all the subgraphs taken from the original graph, that include degree distribution, betweenness centrality, closeness centrality, modularity and clustering coefficient.

1.3 Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2: This chapter introduces the existing approaches of sampling techniques

with their features and strategies.

Chapter 3: Two proposed algorithms, K-Avoiding Random Walk (KARW) and Neighborhood-Avoiding Random Walk (NARW) are described extensively in this chapter with their complexities.

Chapter 4: This chapter includes the simulation of the proposed two algorithms. Simulation environment, parameters and performance metrics have been defined in details.

Chapter 5: Simulation results of the two proposed algorithms based on performance metrics have been collected, analyzed and compared with the existing state-of-the-art algorithms. All of the sampled subgraphs with different variations of K and N in KARW and NARW respectively have been visualized based on key statistical features.

Chapter 6: The conclusion of this thesis has been drawn with final remarks, and future directions are discussed to wrap up the thesis.

Chapter 2

Background and Related Work

2.1 Background

There are some known techniques to measure and analyze interesting key features of large social graphs, such as the betweenness centrality, closeness centrality, degree distribution, number of shortest paths, modularity and clustering coefficient and so on. Due to the massive size and dynamicity of large graphs with billions of users, it is difficult and time consuming to measure these metrics overall. As a result, sampling large graph is essentially required to effectively and efficiently obtain data and analyze its features. In order to estimate these features with sampling, the questions that we need to answer include: (i) which sampling methods are appropriate? (ii) what is the proper size of samples? (iii) whether or not the sampling method can be easily performed in large graphs? As a background study, we are going to review several existing sampling methods in this chapter, including the basic idea and properties of each method.

2.2 Classification of Sampling Methods

Most of the real-world complex networks are dense graphs, each subgraph taken by sampling strategies represents a certain community and plays an important role in many application contexts. In order to collect information from online social networks, graph sampling is necessary and can be categorized into several different ways: (a) graph traversal techniques, (b) sampling by explorations, (c) random node selection sampling, and (d) random edge selection sampling.

2.3 Related Work on Graph Traversal-Based Sampling

In the graph traversal technique, each node in the connected component is visited exactly once until all nodes are visited. Graph crawling methods include Breadth-First Search (BFS), Snow-Ball Sampling (SBS), Respondent-Driven sampling, Forest Fire Sampling (FFS) and so on.

2.3.1 Breadth-First Search

Breadth-First Search (BFS) is a basic strategy that has been used comprehensively for sampling OSNs by many researches. Starting from an initial node, in each iteration, BFS samples the next neighbor of the initial node, which has not yet been visited. When all neighbors of the initial node are visited, BFS moves to the first neighbor of the initial node and repeats the same process from there. It has been shown that this method is biased towards nodes with high degrees [9][29][31]. In other words, after a limited number of steps, nodes with higher-degrees tend to be selected with a higher chance. This bias towards higher-degree of nodes has been confirmed in [24] with a measurement of Facebook [17] data, where their [24] BFS crawler found the average degree of nodes to be 324, while the real value was only 94 (i.e., about 3.5 times smaller).

2.3.2 Snowball Sampling

Snowball sampling was developed by Coleman (1958–1959) [15] and Goodman (1961) [19] to study the structure of social networks. It is a non-probabilistic sampling technique that existing subjects (e.g., persons or animals) recruit future subjects from their acquaintances. It is called snowball sampling because similar groups tend to grow larger as the sampling process moves ahead. The main advantage of snowball sampling is that it can help reach hard-to-reach population after several iterations. As such, snowball sampling has been used in many areas, including public health (e.g., drug users), public policy (e.g., illegal immigrants), and arts and culture (e.g., musicians) [22]. Clearly, due to the snowball nature, this method is also biased towards nodes with high degrees. For example, people who have many friends are more likely to be selected into the sample.

2.3.3 Respondent-Driven Sampling

Respondent-driven sampling is a variation of snowball sampling that was introduced by Coleman [15] in 1958, with the goal to avoid bias in snowball sampling. It works in the same way as snowball sampling, but unlike snowball sampling, respondent-driven sampling involves extra custom estimation procedure that identifies and corrects the homophily on attributes in the population [21]. In other words, with the extra custom estimation procedure, snowball sampling tends to select samples from friendship. This type of biasness has been corrected in respondent-driven sampling by avoiding homophily nature of samples.

2.3.4 Forest Fire Sampling

Strictly speaking, forest fire sampling (FFS) is a hybrid method and it combines snowball sampling and random walk sampling [5][6]. Starting from a randomly selected node, FFS "burns" a fraction of its outgoing links. The process is repeated to neighboring nodes just as the spread of forest fire. When no new node can be burned, FFS randomly selects from the graph a new seed node which has not been burned to start the burning process. In other words, the "forest fire" starts from another new region. This process continues until we get the desired sample size.

2.4 Related Work on Random Walk-Based Sampling

In random walk-based sampling, random walks allow each node in the graph to be visited randomly and potentially many times. Random walks have been used extensively for sampling the Web [25], P2P networks [38][18] and other large graphs [38]. The random walk-based methods include Uniform Random Sampling (URS), Metropolis Hastings Random Walk (MHRW), unweighted Random Walk (RW), Self-Avoiding Random Walk (SARW), and so on. Comparing with the graph crawling techniques, random walks have several advantages [34] where it can be used to capture the community structure as well as capturing the structure with overlapping communities that occur in real-world cases. Random walk-based methods are also very helpful in detecting malicious activities of fake users in social graphs [36].

Because of the advantages of using random walk-based method mentioned earlier,

we will mainly focus on random walk-based sampling techniques in this thesis, especially three of the most popular state-of-the-art strategies: Uniform Random Sampling (URS), Random Walk (RW) and Metropolis Hastings Random Walk (MHRW). Nevertheless, to provide a complete picture, we briefly introduce all other existing sampling methods later in this chapter.

2.4.1 Uniform Random Sampling (URS)

In this strategy, nodes of a graph are sampled uniformly [7]. With this sampling technique, subsets of equal size are selected with equal probability. Uniform sampling is considered to be the simplest and most useful sampling techniques. This sampling technique captures the intuitive concept of randomness and the sampled nodes are often considered representative to the whole population [17]. If no information about the entire population is available, uniform sampling is probably the only choice.

2.4.2 Random Walk (RW)

Random walk [7] starts at a random node and advances in each step to a neighbor of the current node at random. The fraction of times that the random walks visit node n after many hops is proportional to the degree of node n, d_n. When the graph is unweighted, the next node the walk moves to, is chosen uniformly at random among the neighbors of the present node [7]. When the graph is weighted, it moves to a neighbor with the probability proportional to the weight of the corresponding edge.

2.4.3 Metropolis Hastings Random Walk (MHRW)

Metropolis Hastings Random Walk (MHRW) [7] avoids bias towards nodes of higher-degrees by modifying the transition probabilities. MHRW works in the two steps: *proposal* for the next move, and *acceptance/rejection* of the proposal. Assume that the current state of the walk is at node i. In the proposal step, the algorithm chooses a node j uniformly at random from the neighbors of node i and proposes to move to j. In the acceptance/rejection step [13], it generates a random number p uniformly distributed in $(0, 1)$. If $p \leq \min\{1, \frac{d_i}{d_j}\}$, where d_i and d_j represent the degree of node i and the degree of node j, respectively, then the proposal is accepted, i.e., the next move of the walk is node j. Otherwise, the walk stays at node i. The

above proposal → acceptance/rejection process repeats again until a desired number of walks is reached.

MHRW suffers from their slow diffusion over the space, which can in turn lead to poor estimation accuracy. In particular, their fully random nature in selecting the next node, when making a transition, often causes the walk to go back to the previous node from where they just came. This behavior may produce many duplicate samples for a short to moderate time span. It is apparently desirable to avoid such backtracking transitions whenever possible, so as to steer them toward "unvisited" places (or to obtain new node samples), as long as such a modification does not affect the unbiased estimation.

2.4.4 Re-Weighted Random Walk (RWRW)

As discussed earlier that random walk is biased towards higher-degree nodes, this technique corrects that problem by assigning re-weighted values. It uses Hansen-Hurwitz estimator [20] to eliminate the problem of bias towards higher-degree nodes in random walks [16]. Hansen and Hurwitz (1943) introduced the notion of sampling unequal size clusters with Probabilities Proportionate to Size (PPS) to estimate X, the sum of the x-variate over a finite population of M elements. Their procedure for sampling clusters with PPS and with replacement has been widely adopted in sample surveys.

2.4.5 Self-Avoiding Random Walk (SARW)

Self-avoiding random walk is defined as a walk along the path of a given graph or a network without any loops, i.e., the walk will avoid any node that has been visited already. This technique has been used extensively for modeling large-scale properties of long-flexible macromolecules in solution, study of polymers and to characterize complex crystal structures [23]. While being useful, SARW may end up with a dead-end path where the walk has nowhere to go.

2.5 Related Work on Random Node Selection Sampling

In this method, nodes in the graph have been sampled based on some criteria and they are classified in the following ways:

2.5.1 Random Node Sampling

A sampled graph with nodes is being created by selecting a set of nodes N, taken uniformly at random and then is induced by set of those N nodes [30]. This algorithm does not retain the power-law degree distribution [37].

2.5.2 Random PageRank Node Sampling

In contrast to uniform sampling, authors in [30] explored a sampling strategy where they set the probability of a node being selected into the sample to be proportional to its PageRank weight. This sampling strategy is called Random PageRank Node (RPN) sampling.

2.5.3 Random Degree Node Sampling

In this sampling technique, nodes with higher-degree are being selected, therefore, Random Degree Node (RDN) sampling is non-uniform and has even more bias towards high-degree nodes [30]. Since too many nodes with high degree would be sampled, it may cause problems when matching the degree distribution and hence no exact information can be found that can represent the original graph completely.

2.6 Related Work on Random Edge Selection Sampling

In this strategy, edges in a graph rather than nodes have been chosen at random. It has been characterized by two different ways:

2.6.1 Random Edge Sampling

Similarly of selecting nodes at random, one can also select edges uniformly at random. This algorithm is referred to as Random Edge (RE) sampling. But RE suffers with drawbacks [30] as sampled graphs taken from the original graph would be connected sporadically and therefore will have large diameters, and will not fall under any community structure.

2.6.2 Random Node-Edge Sampling

A slight variation of random nodes is Random Node-Edge (RNE) sampling [30], where a node is picked uniformly randomly and choose an edge uniformly at random incident to the node. RNE sampling does not have a tendency to bias towards higher-degree nodes.

2.7 Summary

This chapter introduces several popular sampling techniques. The basic steps and the features of each sampling method are presented and explained. Motivated by the self-avoiding random walk, in the following chapters we will propose and evaluate two new sampling strategies, called K-Avoiding Random Walk (KARW) and Neighborhood-Avoiding Random Walk (NARW), to sample social graphs.

Chapter 3

New Sampling Methods

3.1 Motivation

Good sampling methods in large social networks should quickly obtain enough samples that capture the statistical features of the whole network. Sampling methods based on random walk have been shown to be effective but may need a large number of steps to collect enough samples. In some cases, random walks may end up with a dead end. Even in a large network, if the random walks move into a small region that has weak connection to other parts of the network, the random walks may be stuck within the region for a long time, without a chance to sample nodes in the rest of the network. In this case, the samples will lead to a biased view of the whole network. Motivated by the self-avoiding random walk (SARW), in this chapter we propose two new algorithms that help speed up the sampling process and reduce the chance of random walks being limited within small regions.

3.2 K-Avoiding Random Walk

In our first algorithm, we extend self-avoiding random walk to K-Avoiding Random Walk (KARW), where K is a positive integer whose value will determine the maximum number of re-visiting nodes while sampling a graph. The constraint we pose on random walk is that a node should not be visited more than $K-1$ times, avoiding a node K^{th} times. Clearly, when $K = 1$, KARW is the same as the self-avoiding random walk. When $K \to \infty$, KARW is equivalent to the unweighted random walk. For the implementation detail, we show the pseudo-code of KARW here:

Algorithm 1 Node Sampling with K-Avoiding Random Walk

Require: $G, sample\_Size, K$

choose a seed number randomly
define list ka_rw_sample /* An array that allows an element to appear multiple times */
define list currently_Sampled_Neighbors /* A list that only records unique elements */
$starting\_Node = randomly\_choose\_a\_node\_from\_G$
$ka\_rw\_sample = starting\_Node$
$prev\_Node = starting_Node$
while $len(ka\_rw\_sample) < sample\_Size$ **do**
  $currently\_Sampled\_Neighbors = neighbors(G.prev\_Node)$ /* Record all the neighbors of the current node */
  $temp\_Node = rand.choice(currently\_Sampled\_Neighbors)$ /* $temp\_Node$ is the next proposed move */
  while $num\_occurrences(ka\_rw\_sampled, tempNode) > K$ **do**
    if $len(currently\_Sampled\_Neighbors) == 0$ **then**
      print 'Sorry; there has been a deadlock at G.prev_Node because all of its neighbors have been visited k times'
      return ka_rw_sample
      exit;
    else
      remove temp_Node from currently_Sampled_Neighbors /* The proposed move is invalid, update the valid neighbors */
      $temp\_Node = rand.choice(currently\_Sampled\_Neighbors)$ /* Propose another neighbor */
    end if
  end while
  $ka\_rw\_sample.append(temp\_Node)$ /* The proposed move is valid and is added */
  $prev\_node = temp\_Node$ /* Move to the proposed and valid node */
  empty the list of curremtly_Sampled_Neighbors
end while
return ka_rw_sample

This algorithm takes the value of K and sample size which essentially means the intended length of the walk (in hops) as input, returns the sampled nodes visited with the actual length of the walk (as actual hops). The algorithm terminates when it is not possible to move further or finished sampling all the neighbors from the current node. For instance, the walk may terminate prematurely if all the neighbors of the current node have been visited $K-1$ times and there is essentially no node(s) to visit further.

Apart from the above algorithm, we also have developed three additional algorithms that are being used inside and are considered to be a part and parcel of the K-Avoiding Random Walk algorithm. These algorithms include:

- initialize_graph(filename, seed_Val) – This function is used to create a graph, G from a given dataset. We need to generate this graph prior to applying our proposed algorithm.

- display_graph_info(G) – This function is being used to determine the total number of nodes and the total number of edges. It also determines whether the graph is connected or not. If the graph is disconnected, the algorithm returns the number of connected components in the graph.

- num_occurrences(node_list, node) – This function will count the number of occurrence of nodes from a list and returns the exact numbers of presence of nodes in the list. This function is required to check and to make sure each node has been visited a maximum allowable $K-1$ times.

3.3 Neighborhood-Avoiding Random Walk

Our second algorithm is called Neighborhood-Avoiding Random Walk (NARW). The walk starts from a random node in a graph. Depending upon the value of N which determines the level of neighborhood that should be avoided in the walk, the next move is to a node that is outside the N-level neighborhood of the starting node. In other words, NARW moves in a jumping fashion. Note that, node A is called the N-level neighbor of node B if the shortest distance between nodes A and B is N hops. The implementation detail of NARW is shown in the following pseudo-code:

Algorithm 2 Node Sampling with Neighborhood-Avoiding Random Walk

Require: $G, sample\_Size, starting\_Node, seed\_Val, N$

```
choose a seed number randomly
Define list na_rw_sample /* It is an array to hold all the sampled nodes */
neighbor_List = determine_neighbors(G, starting_Node, N) /* List all the neighbors from starting node to Level-N */
skip_Count = 0
max_Nodes = G.number_of_nodes()
while len(na_rw_sample) < sample_Size do
  starting_Node = rand.choice(neighbors(G, starting_Node)) /* Randomly choose a neighbor from starting node */
  if starting_Node not in neighbor_List then
    /* Check if the chosen node is in the neighborhood list */
    na_rw_sample.append(starting_Node) /* Add the node in the neighborhood list */
  else
    skip_Count = skip_Count + 1 /* Keeps track of how many nodes are skipped */
    if skip_Count > max_Nodes then
      print 'There are no nodes left to be sampled'
      return na_rw_sample list
      exit;
    end if
  end if
end while
return na_rw_sample list
```

The above algorithm takes the value of N which represents the maximum level of neighborhood that needs to be avoided and sample size which determines the intended length of the walk (in hops) as input. Starting from the first node chosen randomly, the next move "jumps" to one of the current node's N-level neighbors and samples nodes thereafter. The algorithm terminates when from the current node, there is no N-level neighbor remain unvisited.

For the graph initialization and the display of graph information, we have used two similar functions mentioned in the previous section: initialize_graph( ) and dis-

play_graph_info( ). Moreover, there is an additional function associated with NARW, which is briefly described as follows:

- determine_neighbors(G, Node, N)– This algorithm is being used to determine neighbors of N-level from the starting node, chosen randomly.

3.4 Complexity of the Proposed Algorithms

Since at each step both algorithms only need to check the local information of the current node (direct neighbors for KARW or N-level neighbors for NARW), the complexities of both KARW and NARW are $\mathcal{O}(n)$, where n is the total number of samples.

3.5 Summary

In this chapter, we have proposed two random-walk based methods for sampling nodes in a large graph. The motivation of both algorithms is to avoid re-visiting a node multiple times or avoiding local regions and to "push" the walk quickly across the network. While behaviors of these sampling methods are straightforward, their performances are not being analyzed mathematically. To this end, we perform comprehensive simulation study to evaluate the performance of KARW and NARW by sampling a large-scale social network in the following chapter.

Chapter 4

Simulation and Performance Metrics

4.1 Introduction

In this chapter, we evaluate our proposed sampling algorithms named K-Avoiding Random Walk (KARW) and Neighborhood-Avoiding Random Walk (NARW). We compare these two algorithms with existing state-of-the-art algorithms: 1) Uniform Random Sampling (URS); 2) Random Walk (RW); and 3) Metropolis Hastings Random Walk (MHRW). We setup our simulation environment and implemented all the five algorithms using Python [35]. The algorithms were evaluated on a large publicly available real-life social network dataset (Mislove [41]) of Facebook user interactions. The simulation results from all algorithms are then statistically analyzed and visualized using Gephi [8]. The analysis results are shown in various tables while the visual comparisons are shown using graphs in the following chapter. The experimental setup, simulation environment, simulation parameters, the dataset, and performance metrics are explained in the following subsections:

4.1.1 Experiment Setup

The simulation was performed on a Linux (Ubuntu 14.04) machine with a quad-core Intel Xeon E5420 CPU and 8 GB of RAM. Different scenarios were created by changing the simulation parameters. The five algorithms were then compared for each scenario using the Facebook dataset. In many realistic scenarios, our proposed algorithms perform better than the other existing techniques.

4.1.2 Our Simulation Tool

We implemented a simulation tool in Python [35] to evaluate our algorithms. The simulation tool is independent and could be used for all evaluation scenarios by changing the performance parameters. The existing and the proposed algorithms are added as separate Python [35] modules to the tool. The advantage of implementing all the algorithms and simulation tool using single platform/programming language gives us the ease of its use.

4.1.3 The Dataset

We have studied and used a large publicly available social network dataset (Mislove et al [41]), collected from the most popular social network, Facebook. This dataset consists of user links and their wall posts from the New Orleans regional network with a total of 63,000 users and about 1 million user-to-user links. The dataset was collected for the duration of around 3 years. For further details on the dataset, and for its statistical features, we refer the interested readers to Mislove et al [41].

4.1.4 Simulation Parameters

The simulation parameters used in our simulation are described below:

- **Maximum number of re-visiting nodes K:** The parameter K limits the re-visiting of a node. We test our algorithms for $K = 1$ to $K = 32$ incrementing it according to Geometric sequence. For $K = 1$, the algorithm will not allow any re-visit to a node i.e., a node can be visited only once at maximum. For $K = 32$, a node could be visited 32 times (31 re-visits) at maximum.

- **Level of neighborhood N:** The parameter N represents how many levels of neighborhood nodes will be avoided during the walk when we apply NARW algorithm to sample original graph. This algorithm samples nodes by pushing outside of the N^{th} level of the neighborhood, starting walk from a node, chosen randomly. The value of N can be any positive number but for our simulation purpose we have used 1, 2, and 4. The algorithm terminates prematurely at $N = 4$ without completing the intended length of the walk when the number of neighborhood nodes becomes very high as the neighborhood size becomes 57K depicted in Table 5.2 in the next chapter, however there are 63K nodes in the

original graph and therefore there would be only 6K nodes left to be visited following this technique.

- **Sample size/initial length of the walk L_w:** The initial or intended length of the walk (L_w) is the size of the sample. We use $L_w = 5000$ to 1.28×10^6 incrementing each time by geometric series i.e., $5000, 10000, 20000, \ldots, 1.28 \times 10^6$. For smaller values of K (and also larger values of N), the re-visits in the walks are restricted and both the algorithms terminate without reaching to the intended length of the walk, L_w.

4.1.5 Performance Metrics

The performance metrics to compare our proposed algorithms with the existing solutions are described as follows:

- **Length of the walk completed:** This metric explains the actual length of the walk completed for KARWs and NARWs for each entry of the intended length of the walk. For other current sampling algorithms, actual length and intended length were same, but in our two proposed algorithms, restriction in re-visiting nodes for small values of K in KARWs and lack of nodes outside of the neighborhood in NARWs to visit for larger values of N are not allowing walks to finish as they were intended. We are measuring the performance as how far (by hops) a walk can end up finishing by each algorithm as compared to what was targeted before starting the walk.

$$Length\_of\_the\_walk\_completed(\%) = \left(\frac{p}{q}\right) \times 100 \tag{4.1}$$

where p = *Actual length completed by a walk and*

q = *Intended length of the walk*

- **Number of nodes sampled over entire population:** The metric here explains the percentages of the number of nodes that are sampled from the entire population (63K nodes) during a walk with different lengths of intended walk using existing and proposed algorithms. This value has been taken as ratio with the entire population of the main graph and is being derived in the following

equation:

$$nodes\_sampled\_over\_entire\_population(\%) = \left(\frac{y}{x}\right) \times 100 \tag{4.2}$$

Where, x *= Total number of nodes in the main graph,*
y *= Total number of sampled nodes and*

- **Number of nodes sampled exactly once:** This performance metric is the ratio between the total number of nodes that are sampled exactly once and the total number of nodes sampled during a walk. For KARWs, the intention is to avoid re-visits by restricting values of K. Due to the restriction in KARW, it may not be able to sample more nodes than expected, however, there is a likelihood of more nodes sampled uniquely than the existing state-of-the-art algorithms. For example, when K=1 in KARW, there is no chance of re-visits, as a result, it will eventually sample every single nodes exactly once. For NARWs, this algorithm tends to push its locality and jumps to a different region, skipping level-N of neighborhood nodes and samples nodes thereafter. Due to less number of nodes at level-N, there is a high probability of getting more uniquely sampled nodes when N=4 in NARW than the other two variations of NARWs.

  Based on the concept discussed here, we have developed an equation that is defined as follows:

$$nodes\_sampled\_exactly\_once(\%) = \left(\frac{z}{y}\right) \times 100 \tag{4.3}$$

  y *= Total number of sampled nodes and*
  z *= Total number of nodes sampled exactly once*

- **Degree Distribution:** The degree of a node in a network is the number of connections it has to other nodes and the degree distribution is the probability distribution of these degrees over the entire network.
  Degree distribution plays a vital role in studying theoretical networks and real networks, such as the Internet and social networks. Networks like the Internet,

the world wide web, and some social networks are found to have degree distributions that approximately follow a Power-law [14]. Networks as such are called scale-free networks and have attracted specific attention for their structural and dynamic nature.

The degree distribution, *D(d)*, of a network is defined in [40] to be a fraction of nodes in the network with a degree d:

$$D(d) = n_d/n \tag{4.4}$$

Where, $n = Total\ number\ of\ nodes\ in\ a\ graph\ and$
$n_d = Number\ of\ nodes\ that\ have\ the\ degree,\ d$

- **Betweenness Centrality:** Betweenness is considered as a measure of the centrality of a node in a network. It is calculated as the fraction of the shortest paths between pairs of nodes that pass through the node of interest. Betweenness is also a measure of the influence of a node in spreading information throughout the network. The measure is based on random walks, counting how often a node is traversed by a random walk between two other nodes [32]. Nodes with betweenness are used to connect different regions in the graph and provide seamless connection from end-to-end nodes even though they do not belong to a same region. The following is the standard measure of centrality:

$$C_B(x) = \sum_{a \neq x \neq b \in V} \frac{\sigma_{ab}(x)}{\sigma_{ab}} \tag{4.5}$$

In the above equation, $C_B(x)$ is the betweenness centrality of a vertex x, $\sigma_{ab}(x)$ denotes the number of shortest paths from node a to node b that some $x \in V$ lies on and σ_{ab} is the total number of shortest paths between a and b.

Betweenness centrality is an interesting metric, because it takes into account not only the local neighborhood for each node, but also the entire graph's structure. Nodes with the highest betweenness centrality are like the junctions in network. The higher this measure is, the more likely it will cross those nodes

when traversing the network. Such nodes may not have the highest degree, but they are very influential, because they tie up different distinct clusters together.

- **Closeness Centrality:** Closeness can be regarded as a measure of how fast information spread from a node to all other nodes in a network [12]. This type of centrality has been defined as:

$$C_C(n) = \frac{1}{\sum_{n \in V} d_G(n, b)} \tag{4.6}$$

  Where $d_G(n, b)$ denotes the distance between vertices, which is the minimum length of any path connecting nodes n and b in a graph G.

- **Modularity:** Modularity is a metric to define the structure of networks and graphs. It is also being used to measure the strength of groups, clusters, and communities within a network. Modularity also confirms the certain level of interactions among nodes in a network. Certain number of nodes that are closely connected form a strong social community which implies information propagation at a faster rate among members within the community. The algorithm for modularity is implemented in Gephi [8] based on [11], that checks step by step the densely connected nodes because nodes that are more densely connected to each other are considered to be belong to the same community. Communities within a network are sparsely connected to one another. Modularity plays a significant role to study many real world problems such as biological and social network phenomenon.

- **Clustering Coefficient:** Clustering coefficient is a measure of how nodes in social networks are tightly knitted to make a cluster. It has been categorized in two ways: the global and the local. Global clustering is a measure of the overall situation of clusters in a network. It is based on the ratio between number of close triplets and number of connected triplets of vertices in a network [43]. Triplets are three nodes that are connected by two or three undirected ties. Local clustering coefficient of a vertex in a graph measures how close the neighbors are to make a complete graph or clique [33].

  As of the definition described above, the average of the local clustering coeffi-

cients of all the vertices n, $\bar{C}$ is:

$$\bar{C} = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{4.7}$$

Where Clustering_Coefficient,

$$C_i = \frac{no.\_of\_connections\_in\_the\_neighborhood\_of\_a\_node}{the\_number\_of\_connections\_if\_the\_neighborhood\_was\_fully\_connected} \tag{4.8}$$

and a fully connected group of n nodes has connections of

$$\frac{n * (n-1)}{2}$$

According to the definition above, the global clustering coefficient is:

$$C_G = \frac{no.\_of\_closed\_triplets}{number\_of\_connected\_triplets\_of\_vertices} \tag{4.9}$$

Triangles in clusters play an important role in complex network analysis. In particular, two prominent theories according to which triangles are generated in social networks are the homophily and the transitivity[42]. Based on homophily nature, people tend to choose friends that are similar to themselves, which is also known as "birds of a feather flock together" and due to transitivity nature, people who have common friends tend to become friends themselves [39].

Chapter 5

Simulation Results and Observations

5.1 Simulation Results

We ran our simulation for all possible and realistic scenarios of the simulation parameters. For KARW, the number of possible scenarios is $sizeof(K) \times sizeof(L_w)$ while for the NARW, it is $sizeof(N) \times sizeof(L_w)$. In our case, the Facebook dataset was exhausted for $sizeof(K) = 6$, $sizeof(N) = 3$, and $sizeof(L_w) = 9$. The performance values for all these scenarios are as follows:

5.1.1 Length of the walk completed

Our first result is based on the length of the walk completed by both algorithms; given a set of intended lengths of the walk (L_w) as input we need to observe the actual lengths of the walk completed by KARW and NARW. We ran our program five times and get the average actual lengths of the walk performed by three different variations of KARW and NARW algorithms. Initial input lengths are set from 5 Khops to a maximum of 1 Mhops, increasing the lengths each time by a factor of 2. All the outputs have been taken with different values of K ($1 \leq K \leq 32$) and N ($1 \leq N \leq 4$) in KARWs and NARWs respectively. We also measure the output for the existing state-of-the-art algorithms and as there is no restriction imposed of re-visiting the nodes while walking on the graph, all of them ended up completing walks as was intended. Lengths of the walk completed by our proposed algorithms are measured with different initial lengths (L_w) and are depicted in Table 5.1. We

have chosen to show the output of actual lengths for three different values of K and N in the following Table 5.1:

Table 5.1: Length of the walk completed by KARWs and NARWs

Initial Length (L_w)	KARW			NARW		
	$K = 1$	$K = 8$	$K = 32$	$N = 1$	$N = 2$	$N = 4$
5000	34	5000	5000	5000	5000	5000
10000	36	10000	10000	10000	10000	335
20000	143	14057	20000	20000	20000	2557
40000	121	7013	12207	40000	40000	3162
80000	288	20784	55482	80000	80000	9792
160000	165	37832	87429	160000	160000	2430
320000	179	51639	89391	320000	305434	1657
640000	153	48392	85744	640000	453766	6297
1280000	220	509348	91369	819200	904194	12665

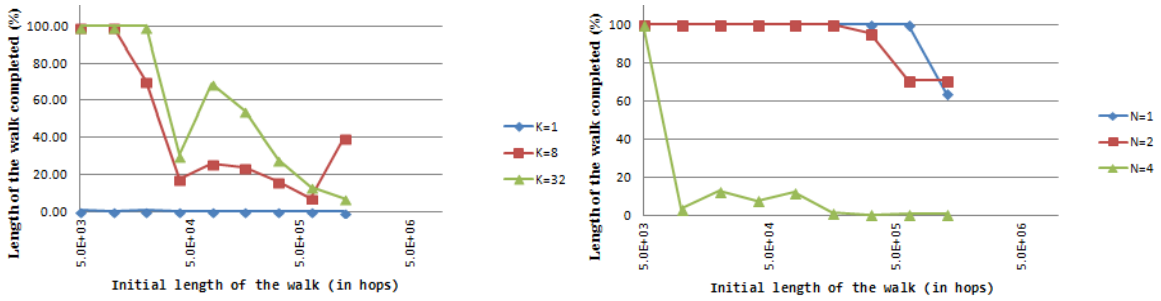
If we closely look at the previous table, we can observe that, for smaller values of K in KARW, this algorithm is not able to complete the walk as intended. The reason behind is very obvious because when the value of K is smaller, then the freedom of movement to re-visit nodes inside the graph is very much restricted. For instance, when $K = 1$, no nodes are allowed to be re-visited while walking on the graph. This constraint has resulted less hops completed by walks when we apply KARW algorithm to sample nodes from an original graph derived from the Facebook dataset. On the other hand, when the values of K in KARW become higher, the more freedom this algorithm has to re-visit nodes in a graph, resulted finishing walks with more hops observed in Table 5.1 when the values of K in KARW are, $K = 8$ and $K = 32$. More details of the actual walks with different values of K and N are depicted in Table A.1 with detail results for interested researchers.

In case of NARWs, there are no such restrictions of re-visiting nodes as opposed to KARWs. As a result, NARW performs better than KARW in finishing walks with more hops. However, there is a case where the actual lengths have been reduced significantly when the value of N is increased to 4, observed in the Table 5.1. The reason behind finishing walks with less hops is, NARW algorithm pushes itself to a region outside of the neighborhood of level-4 and starts walking from there. If we

look at the Table 5.2, we can see that an average neighborhood size of each level of the original graph grows exponentially. As a result, the size of neighborhood of level-4 in NARW grows very quickly and has become 57K, leaving behind less nodes to be visited as we know the total number of nodes in the original graph is 63K. Actual lengths completed by the three variations of NARW are shown in Figure 5.1b. Moreover, details of the walks by NARWs have been summarized in Table A.2 for interested readers.

Table 5.2: Average neighborhood size of the graph in different levels of NARWs

Level	Average Neighborhood Size
$Level-1$ $(N=1)$	29
$Level-2$ $(N=2)$	1000
$Level-4$ $(N=4)$	57043



(a) KARW (b) NARW

Figure 5.1: Length completed with variations of KARWs and NARWs

Comparison of the actual lengths completed by three different values of K and N in KARW and NARW respectively have been shown by graphs in Figure 5.1. We can see that the performance of NARWs overall is much better than KARWs as NARW has more freedom of re-visiting nodes. Both algorithms suffer when $K = 1$ and $N = 4$ in KARW and NARW respectively, where KARW has no way to re-visit nodes in the first place and in the second place, there are not enough nodes outside of the neighborhood that NARW can complete as intended.

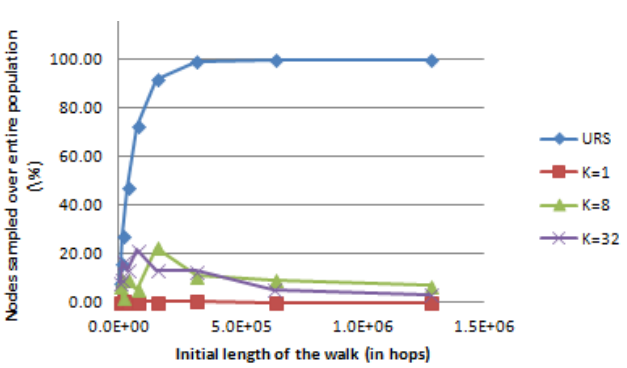
5.1.2 Number of nodes sampled over entire population

Our next simulation result is based on the number of nodes sampled by our proposed algorithms and then compare results with the existing state-of-the-art sampling techniques in different intended lengths. A comparison by percentages of nodes sampled over the entire population between the existing sampling techniques and our proposed algorithms with three different variations of each algorithm is depicted in the following Table 5.3. Interested researchers can consult more details of the results, summarized in Table A.3 and Table A.4.

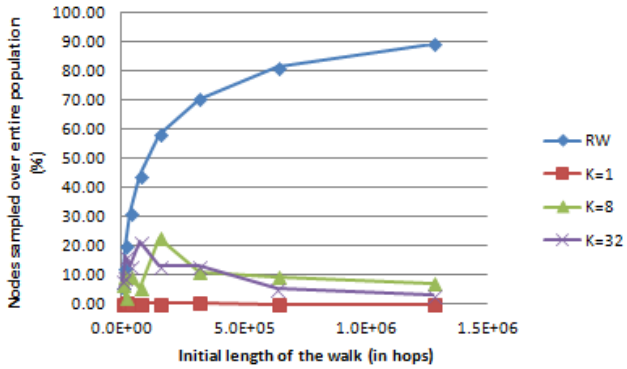
Table 5.3: Nodes sampled over entire population(%)

L_w	URS	RW	MHRW	KARW			NARW		
				$K=1$	$K=8$	$K=32$	$N=1$	$N=2$	$N=4$
5000	8	5.73	3.1	0.054	6.7	8.26	7.9	7.65	6.87
10000	16	12	11.65	0.057	9.04	9.03	11.97	11.98	2.41
20000	27.43	20	11	0.7	2.1	16.37	20	19.89	2.85
40000	47.34	31.23	20	0.3	9.03	13.28	31.74	30.68	2.73
80000	72.65	44.17	34	0.3	5.28	21.46	43.54	43.57	8.77
160000	92	58.35	52	0.1	22.58	13.27	58.3	56.39	2.67
320000	99	70.45	70	0.5	10.76	12.87	70.12	67.84	1.95
640000	100	81.32	82	0.02	9.06	5.3	80.43	72.59	4.78
1280000	100	89.45	90	0.01	6.75	3.09	88.75	80.675	6.74

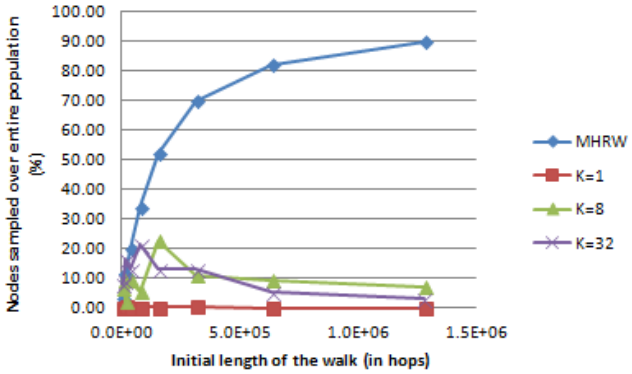
If we consult Table 5.3, we can see that in case of KARWs, there are less nodes sampled due to restriction of movements. As a result, the percentages of nodes sampled are the lowest when $K = 1$ in KARW. As the value of K is increased, the number of sampled nodes is increased. The best result among these three variations can be obtained when $K = 8$ and is measured 22.58%. The rest of the values are lower than 22.58%, meaning KARW algorithm has less freedom of movements, resulted very small amount of nodes being sampled when values of K are smaller. Among existing sampling techniques, URS performs best with sampling more nodes followed by RW and MHRW with a maximum of 90% nodes sampled for both cases. More details of the results of KARWs can be found in Table A.3. Comparison graphs between these three existing techniques with our proposed KARWs have been depicted in Figure 5.2.



(a) KARWs vs URS



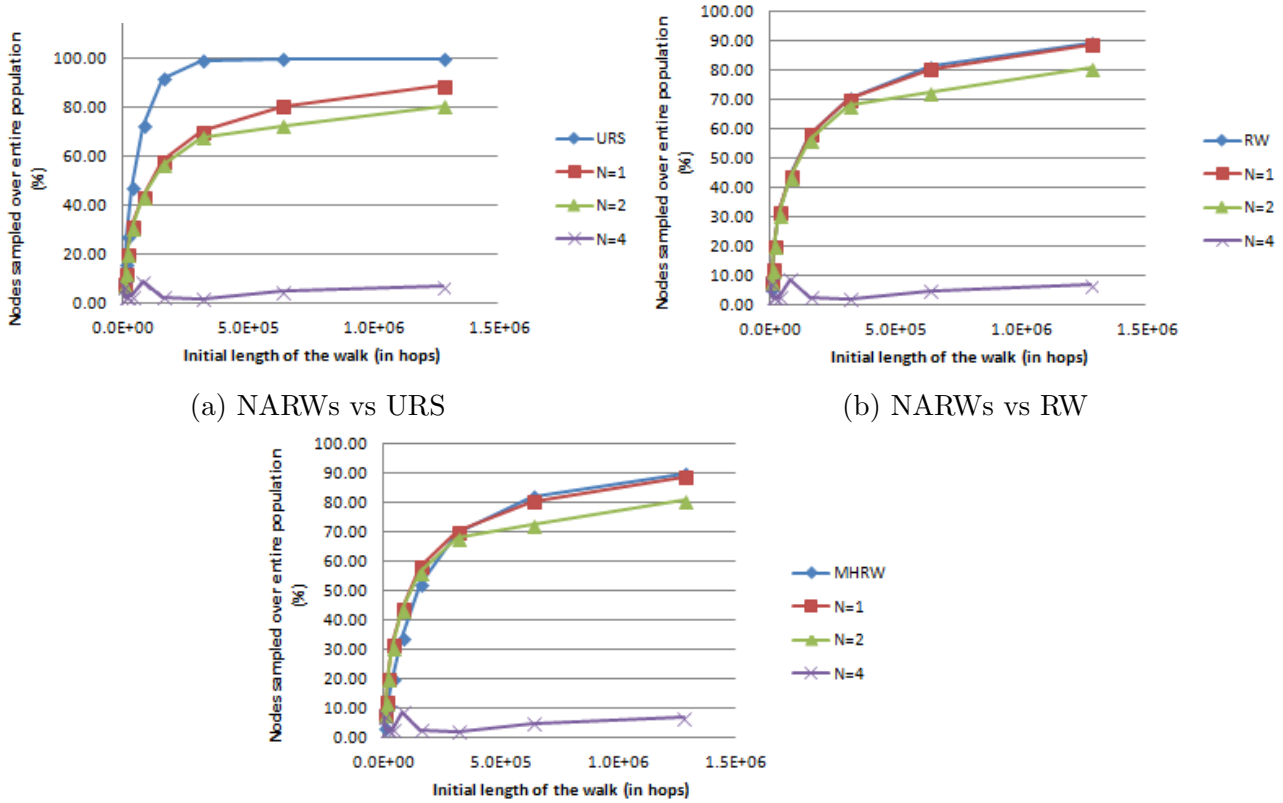
(b) KARWs vs RW



(c) KARWs vs MHRW

Figure 5.2: KARWs vs (URS, RW and MHRW) by nodes sampled over entire population (%)

In case of NARWs, simulation results based on the number of nodes being sampled from the entire population with different values of N are illustrated in Table 5.3 and are much promising when values of N are 1 and 2, and the maximum number of nodes sampled are 80.75% and 80.675% respectively of the entire population. However, lack of nodes outside of the neighborhood when $N = 4$, causes declining the rate of sampling nodes overall with a maximum of 8.77% of the nodes being sampled from the entire population. More details can be found in Table A.4 for interested readers. Comparison graphs between NARWs with the existing sampling strategies have been depicted in Figure 5.3. It has been observed that the results of NARW with $N = 1$ and $N = 2$ are very close to RW and MHRW techniques whereas URS dominates all the way through over other algorithms especially when the initial lengths (L_w) are higher.



(a) NARWs vs URS

(b) NARWs vs RW

(c) NARWs vs MHRW

Figure 5.3: NARWs vs (URS, RW and MHRW) by nodes sampled over entire population (%)

5.1.3 Number of nodes sampled exactly once

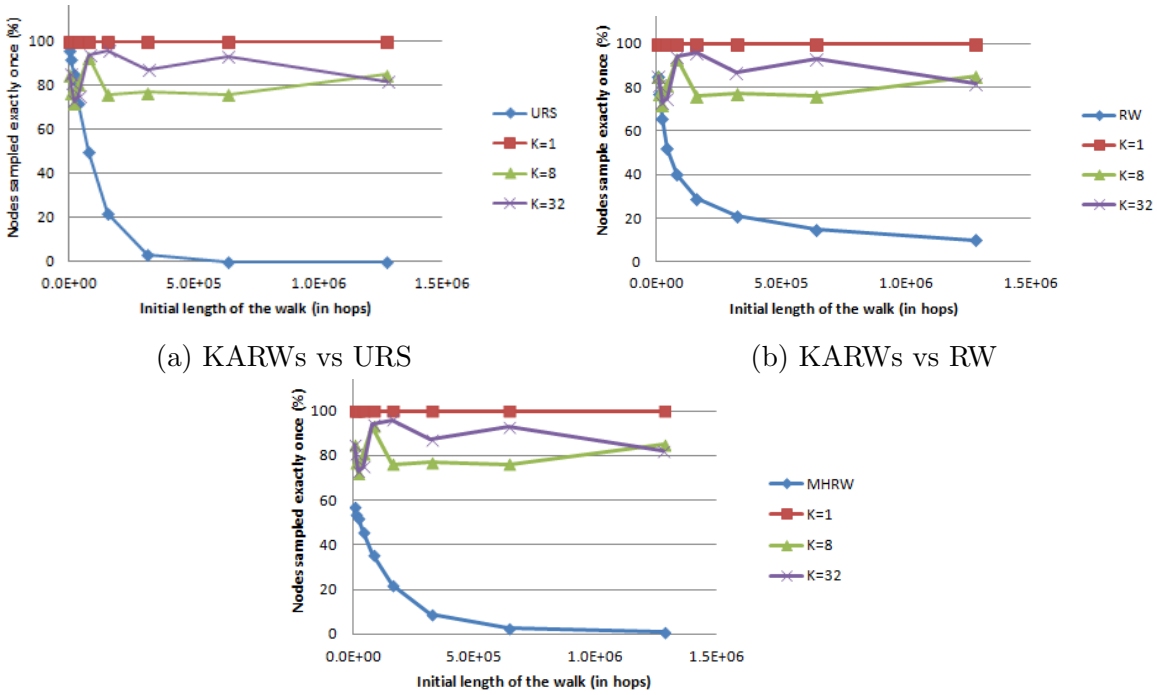
It is interesting to see that how many nodes are sampled uniquely among the total number of sampled nodes during a walk. For example, think about a real situation where a company may want to introduce a new product online. Before it will be introduced to the entire online users, a user-base consisted of potential users has been chosen for their valuable feedback who essentially represent the entire online customers. If random sampling techniques choose similar set of people every time, it causes misleading feedback rather than constructive because of the flaw of sampling similar people time and time again. By introducing two proposed algorithms, we try to reduce the rate of choosing nodes multiple times as well as increasing the rate of uniquely sampled nodes over the other state-of-the-art techniques. The following Table 5.4 shows comparative results of existing strategies with our two proposed techniques with different values of K, N and initial lengths (L_w). More details of

simulation results are illustrated in Table A.5 and Table A.6 for interested researchers.

Table 5.4: Nodes sampled exactly once (%)

L_w	URS	RW	MHRW	KARW			NARW		
				$K=1$	$K=8$	$K=32$	$N=1$	$N=2$	$N=4$
5000	96	85	57	100	85	85	84	83	78
10000	92	77	54	100	77	81	77	76	74
20000	85	66	52	100	72	73	66	65	74
40000	72	52	46	100	81	75	52	52	68
80000	50	40	36	100	93	94	40	40	61
160000	22	29	22	100	76	96	29	29	73
320000	3	21	9	100	77	87	21	22	74
640000	0	15	3	100	76	93	15	18	56
1280000	0	10	1	100	85	82	10	13	46

It has been seen from Table 5.4 that $K = 1$ in KARW performs best among all the other techniques with a 100% uniquely sampled nodes among total number of nodes being sampled. The other two variations of KARW have carried out sampling nodes uniquely within a range of 72% to 96%. The most compelling evidence of KARW is, even though increasing value of K gives more freedom to re-visit nodes, the sampling tends to get evenly distributed. Among the state-of-the-art strategies, there are fewer nodes left to be sampled uniquely than the proposed KARW and NARW algorithms because when the lengths of the intended walk are increased, there is a likelihood of re-visiting nodes a number of times rather than visiting nodes exactly once. Comparison between the existing strategies and the proposed KARW technique is depicted in the following Figure 5.4 and found out that the KARW algorithm with variations have consistency of visiting more unique nodes than the existing state-of-the-art sampling algorithms.



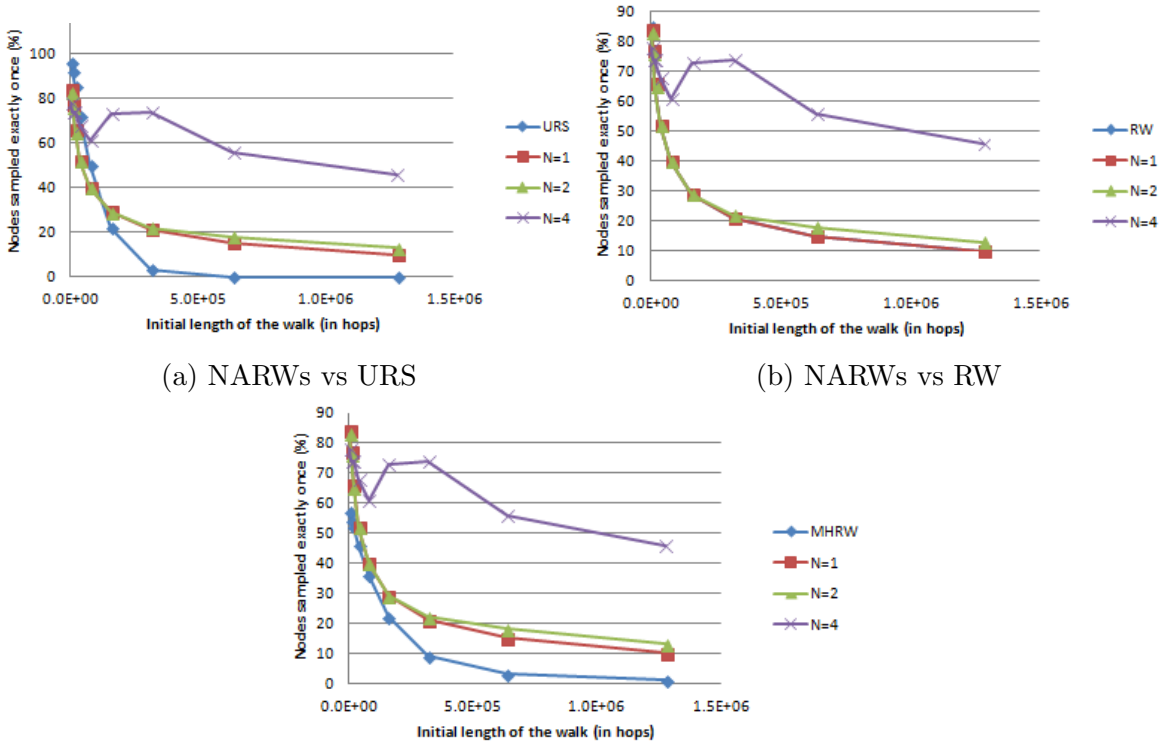
(a) KARWs vs URS

(b) KARWs vs RW

(c) KARWs vs MHRW

Figure 5.4: KARWs vs (URS, RW and MHRW) by nodes sampled exactly once (%)

For our second algorithm, NARW, if we consult Table 5.4, we observe that when $N = 1$ and $N = 2$ in NARW, the rate of uniquely sampled nodes are inversely proportional to the initial lengths of the walk (L_w) with a maximum rate of sampling 84% and 83% unique nodes respectively. When the size of the neighborhood grows to $N = 4$, the results are very consistent as compared to the other two variations of NARW, illustrated in Figure 5.5.



(a) NARWs vs URS

(b) NARWs vs RW

(c) NARWs vs MHRW

Figure 5.5: NARWs vs (URS, RW and MHRW) by nodes sampled exactly once (%)

5.1.4 Degree Distribution

To obtain the degree distribution of KARWs and NARWs, we have used social network analysis tool called Gephi [8] and append all the subgraphs with three different variations of KARWs and NARWs and draw two separate graphs for each proposed algorithm. Afterwards, we calculate the degree distribution for each algorithm and illustrated in Figure 5.7 and Figure 5.8 and observe with the degree distribution of the main graph in Figure 5.6. It has been observed here that all the graphs obtained by different values of K and N in KARW and NARW respectively, have the property of scale-free networks which follow Power-law [14] distribution.

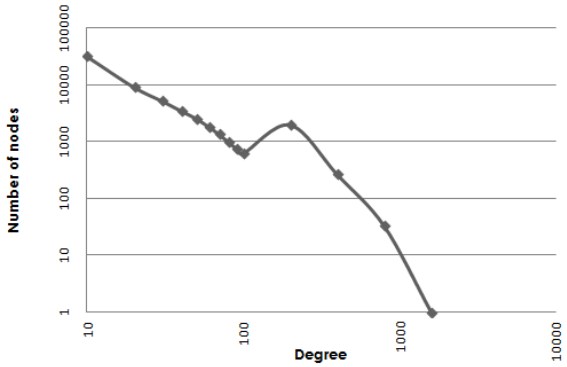
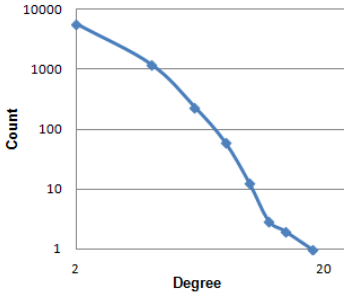
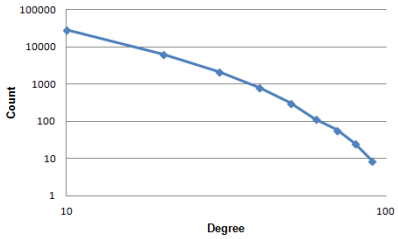


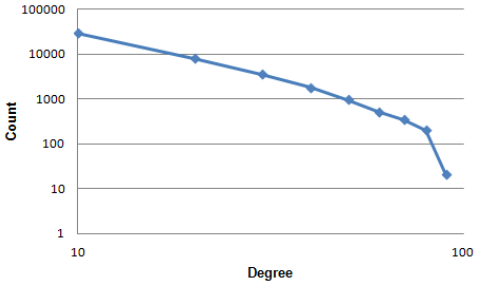
Figure 5.6: Degree Distribution of the main graph (Log-Log plot).



(a) KARW with 5 Khops walk.

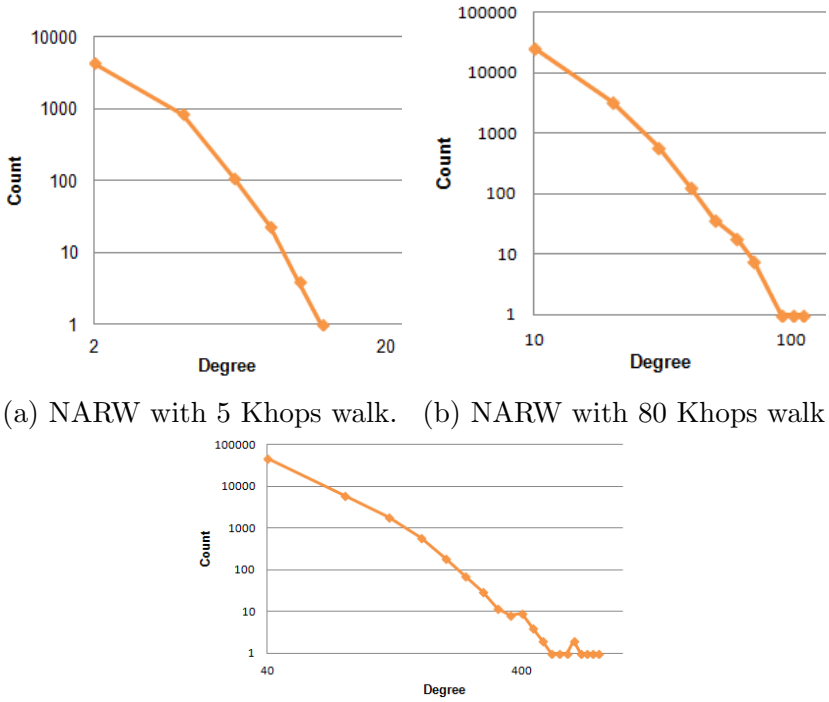


(b) KARW with 80 Khops walk.



(c) KARW with 1 Mhops walk.

Figure 5.7: Degree distribution of KARWs (Log-log plot).

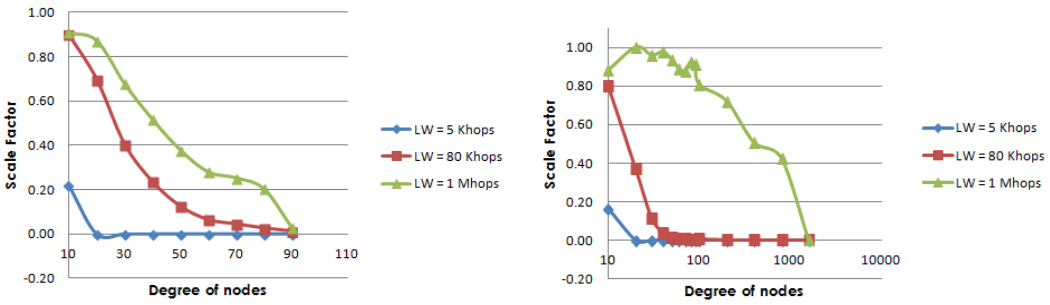


(a) NARW with 5 Khops walk. (b) NARW with 80 Khops walk

(c) NARW with 1 Mhops walk.

Figure 5.8: Degree distribution of NARWs (Log-log plot).

Now we can show how many nodes are there for each range of degrees of nodes and calculate the scale down factor for both KARWs and NARWs with the original graph. Here is the summary of the observation that we find in Figure 5.9. It has been seen here that when the lengths of the walk (L_w) were over a million hops then the scaling factor of both KARW and NARW are higher than the other two versions of the lengths of the walk (L_w) which means the representation of the main graph is close when the lengths of the walk (L_w) are over a million hops.



(a) Scale factor of KARWs. (b) Scale factor of NARWs

Figure 5.9: Scale factor of two proposed algorithms with the original graph.

5.1.5 Betweenness Centrality

The betweenness centrality of the main graph is depicted in Figure 5.10 where we can see there are significant amount of higher betweenness nodes marked as deeper blue colors. In order to observe the betweenness centrality of the subgraphs consisting of sampled nodes, we have used a popular social network analysis tool, called Gephi [8] that accepts sampled data and graphically represents subgraphs with all the central nodes involved in the network, marked by blue-color shades. Higher values of K mean more freedom of re-visiting nodes and we see betweenness nodes tend to be at the center of the sampled graphs. Higher betweenness nodes in the subgraphs have deeper shades of blue-color. Nodes with the highest betweenness have the deepest blue-color and shown in the following graphical representations. For the simplicity of representing graphs, we have taken three different variations of initial lengths (5 Khops, 80 Khops and 1 Mhops) for all possible combinations of K and N in KARW and NARW algorithms respectively, illustrated from Figure 5.11 – Figure 5.16. The other graphical representations for different initial lengths of the walk (L_w) have been depicted in the appendix section.

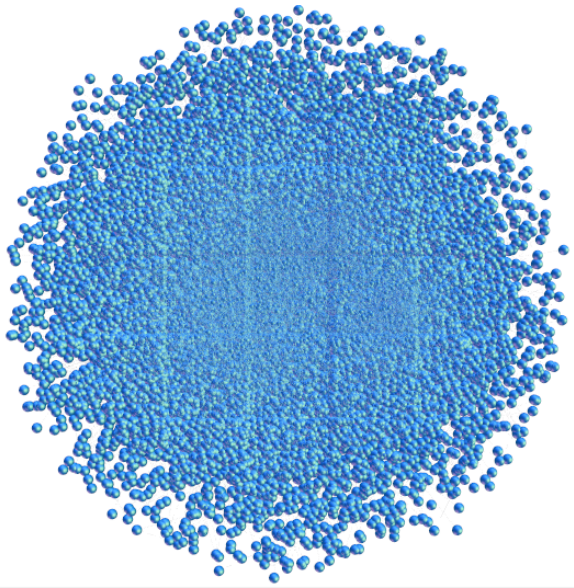
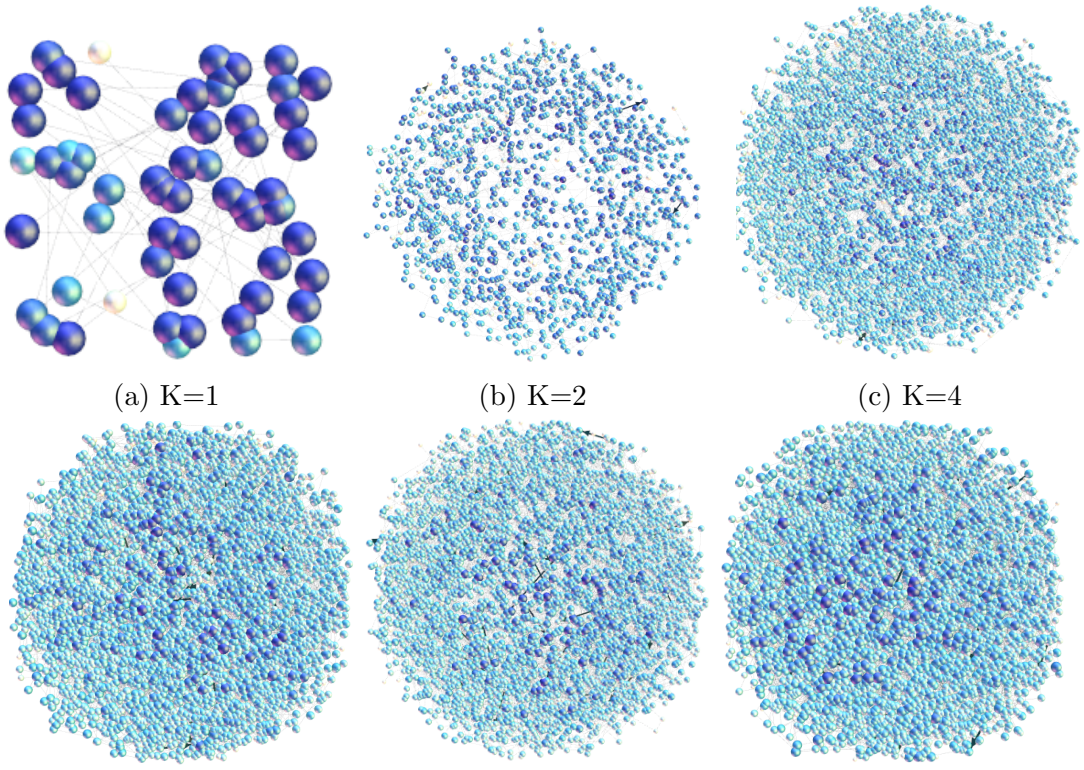


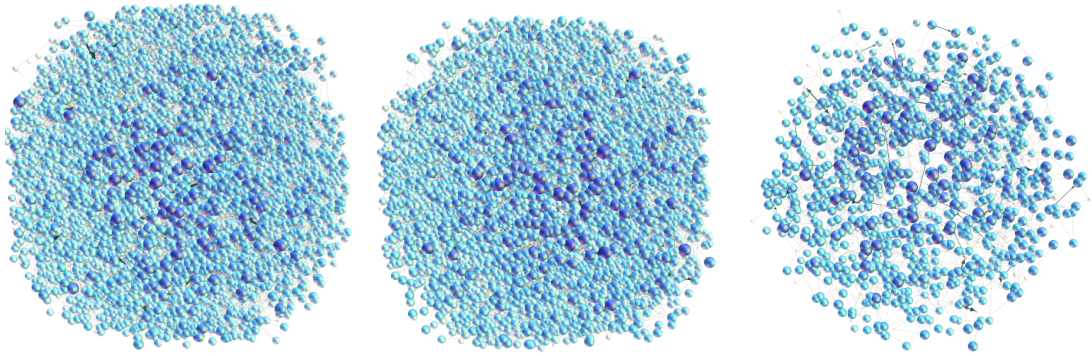
Figure 5.10: Betweeness centrality of the main graph.



(a) K=1 (b) K=2 (c) K=4

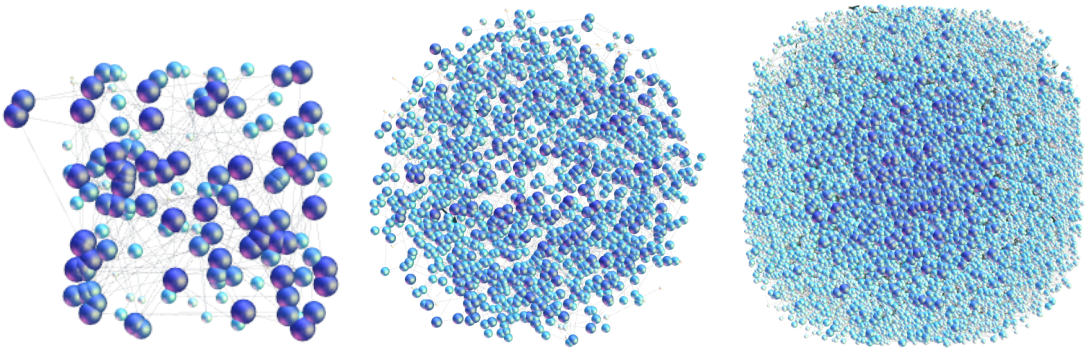
(d) K=8 (e) K=16 (f) K=32

Figure 5.11: Betweenness Centrality of sampled graphs for KARWs with 5 Khops walk

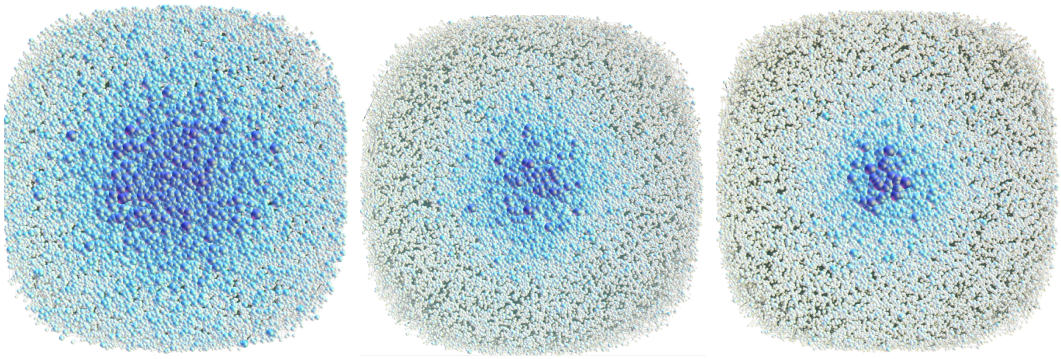


(a) N=1 (b) N=2 (c) N=4

Figure 5.12: Betweenness Centrality of sampled graphs for NARWs with 5 Khops walk

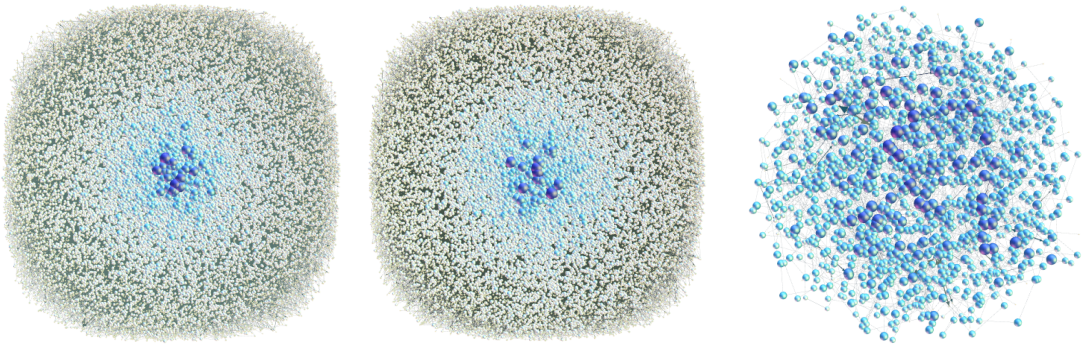


(a) K=1 (b) K=2 (c) K=4



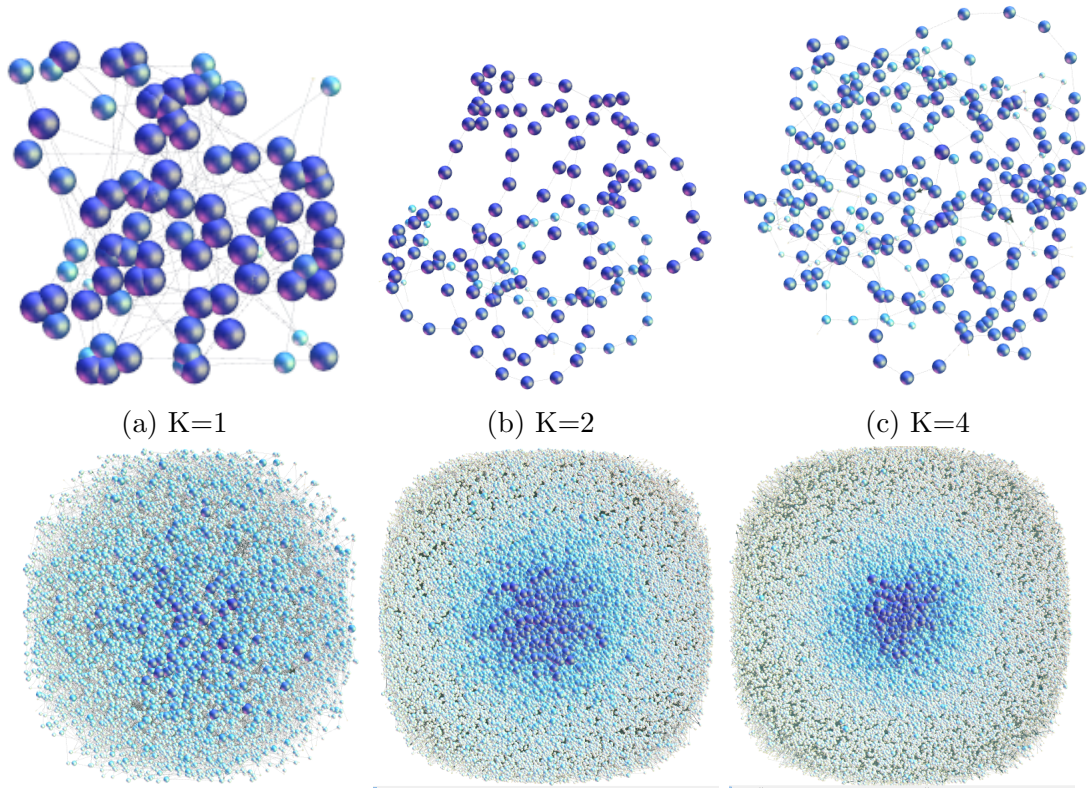
(d) K=8 (e) K=16 (f) K=32

Figure 5.13: Betweenness Centrality of sampled graphs for KARWs with 80 Khops walk



(a) N=1 (b) N=2 (c) N=4

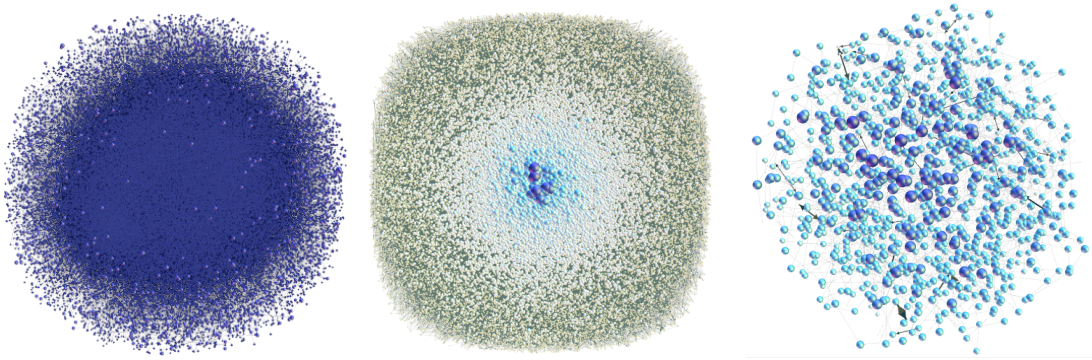
Figure 5.14: Betweenness Centrality of sampled graphs for NARWs with 80 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

Figure 5.15: Betweenness Centrality of sampled graphs for KARWs with 1 Mhops walk



(a) N=1 (b) N=2 (c) N=4

Figure 5.16: Betweenness Centrality of sampled graphs for NARWs with 1 Mhops walk

5.1.6 Closeness Centrality

Closeness centrality measures how fast information spreads over a network from a specific node. Higher closeness nodes are determined and marked with deeper shades of blue-color whereas nodes with lighter shades mean closeness between nodes are lower. For the main graph, the number of nodes with least closeness centrality are seen in Figure 5.17. It has been observed that fewer nodes have higher closeness when the intended lengths of the walk are 5 Khops and 80 Khops. For KARW, when the value of K is 1 which means re-visiting a node is prohibited, the number of closeness centrality nodes become least. As the value of K in KARW is increased, the closeness between nodes increases which means nodes tend to get closer each other causing closeness between nodes gets higher as observed in the following graphical representations. For NARWs, even though the walk starts avoiding neighbors of a certain level, the closeness centrality of graphs tends to get higher when avoiding first and second level of neighborhood nodes.

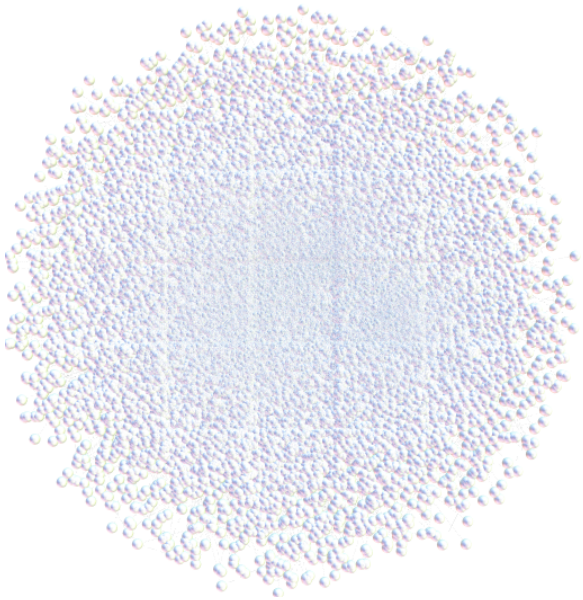
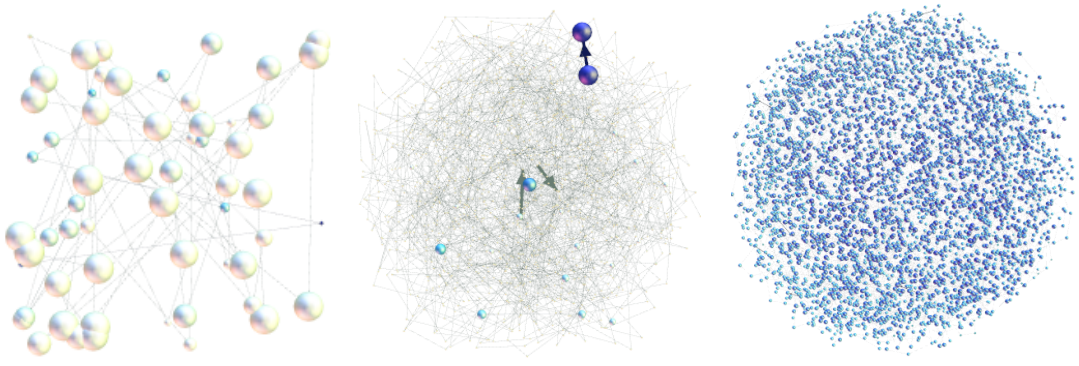
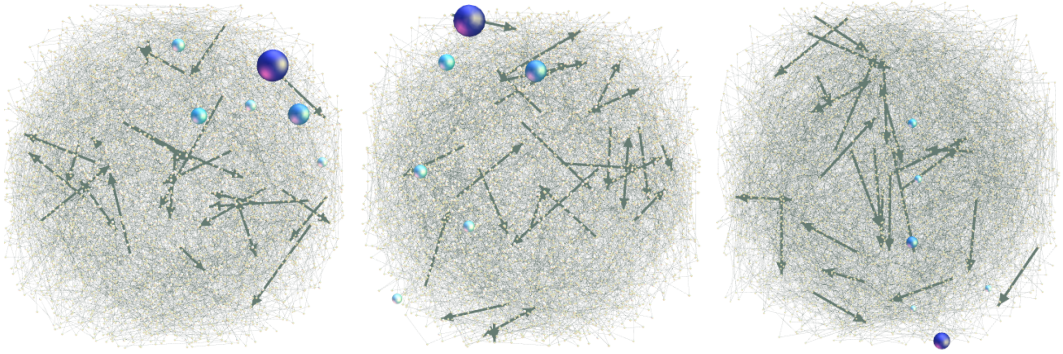


Figure 5.17: Closeness centrality of the main graph.

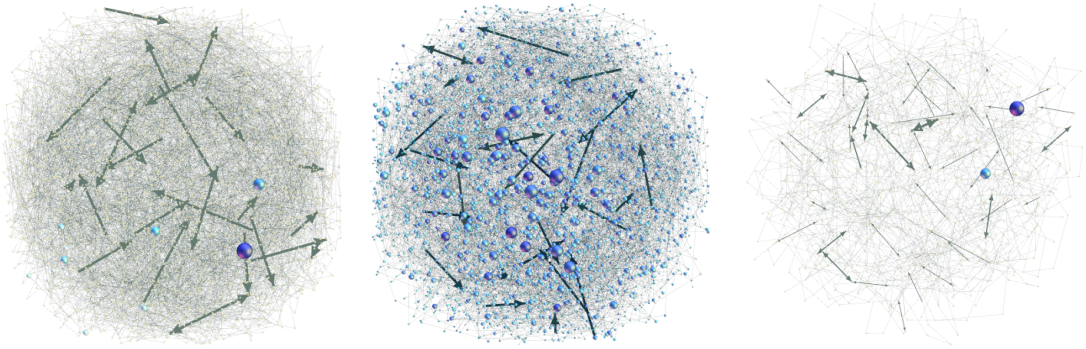


(a) K=1 (b) K=2 (c) K=4



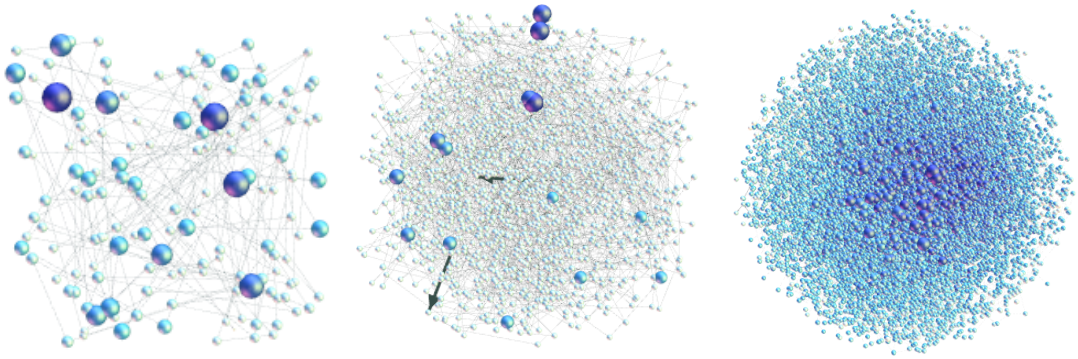
(d) K=8 (e) K=16 (f) K=32

Figure 5.18: Closeness Centrality of sampled graphs for KARWs with 5 Khops walk

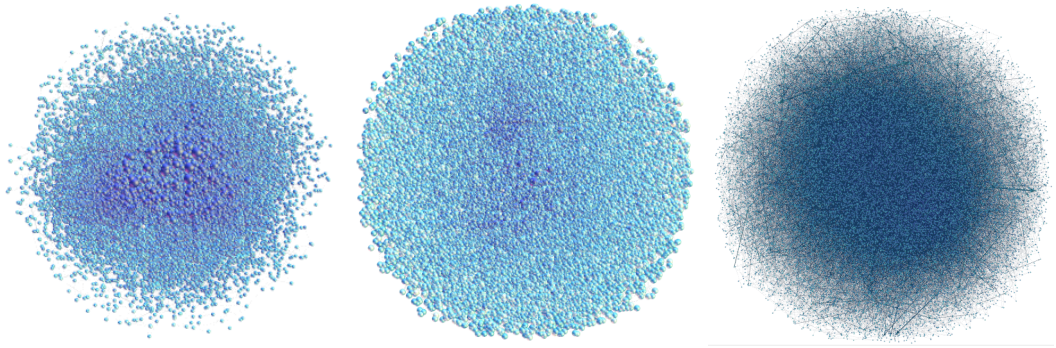


(a) N=1 (b) N=2 (c) N=4

Figure 5.19: Closeness Centrality of sampled graphs for NARWs with 5 Khops walk

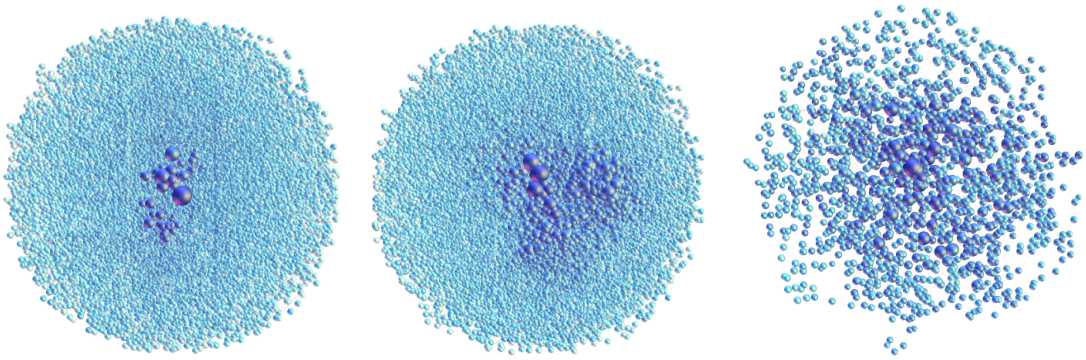


(a) K=1 (b) K=2 (c) K=4



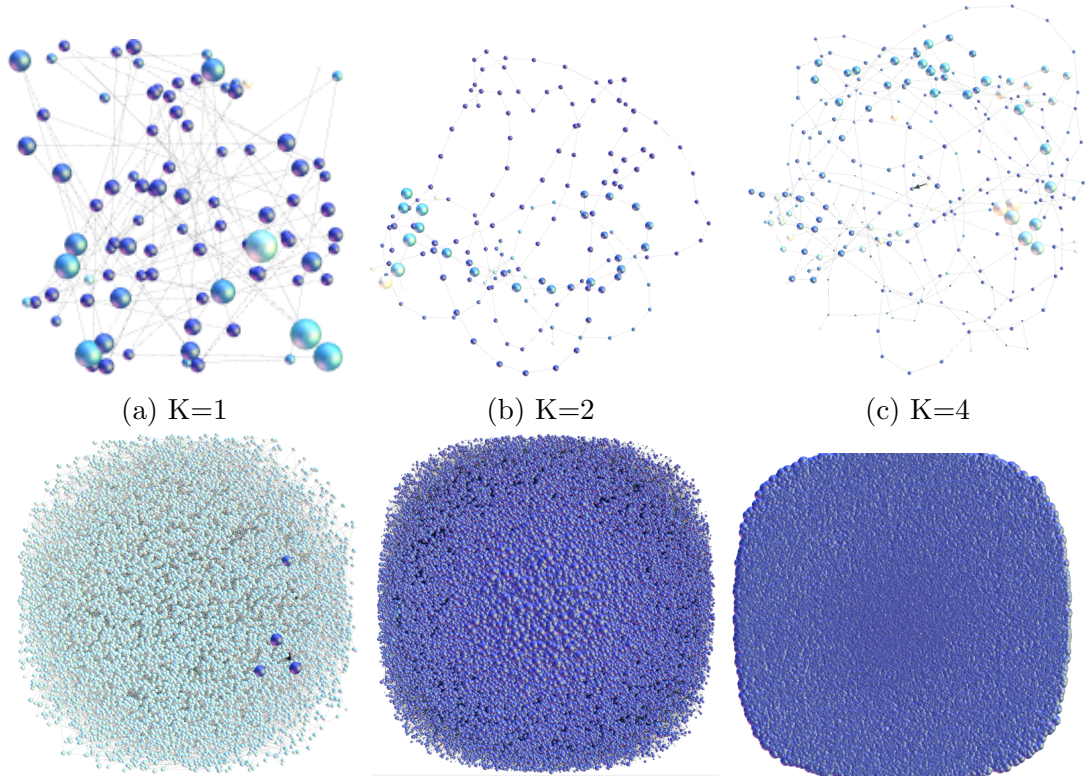
(d) K=8 (e) K=16 (f) K=32

Figure 5.20: Closeness Centrality of sampled graphs for KARWs with 80 Khops walk



(a) N=1 (b) N=2 (c) N=4

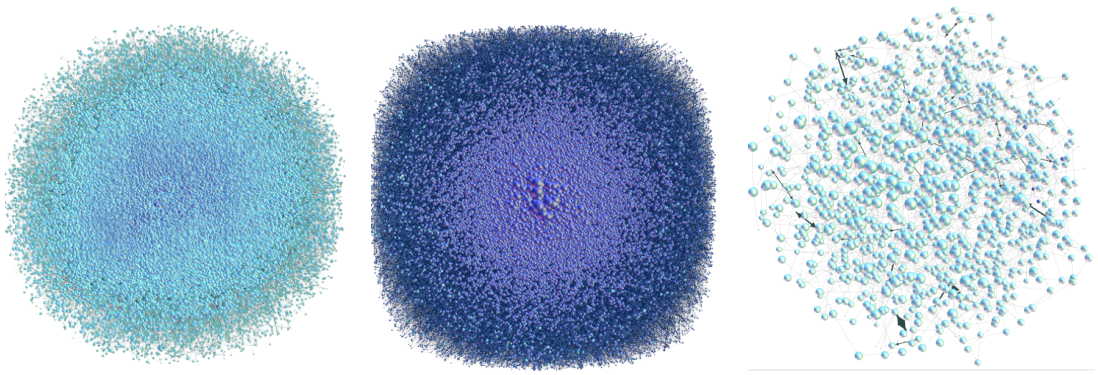
Figure 5.21: Closeness Centrality of sampled graphs for NARWs with 80 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

Figure 5.22: Closeness Centrality of sampled graphs for KARWs with 1 Mhops walk



(a) N=1 (b) N=2 (c) N=4

Figure 5.23: Closeness Centrality of sampled graphs for NARWs with 1 Mhops walk

5.1.7 Modularity

It has been observed from the main graph in Figure 5.24 that a single community in the main graph has been dominated and the size of the other communities are

very small. For observing number of detected communities in sampled subgraphs, a specific algorithm [11] has been used in Gephi [8] for community detection and resolution; it also uses an algorithm mentioned in [27] to produce decomposition. We set resolution = 5 and choose a randomized option in this analysis tool, to produce a better decomposition and apply to all the sampled graphs and observe number of communities detected for both of our proposed algorithms depicted in Table 5.5.

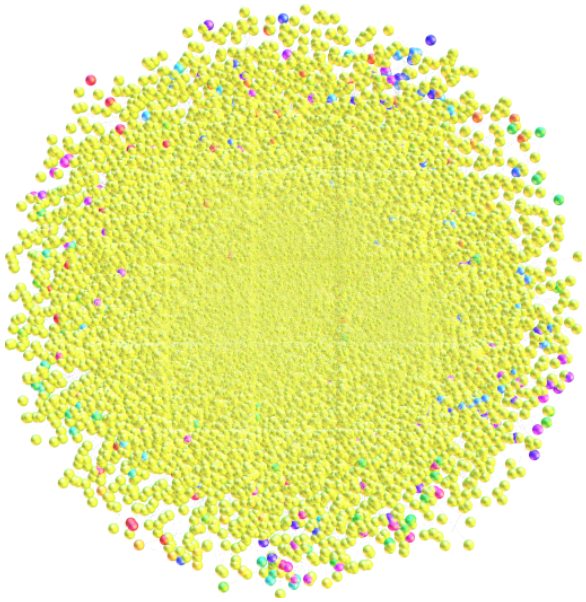
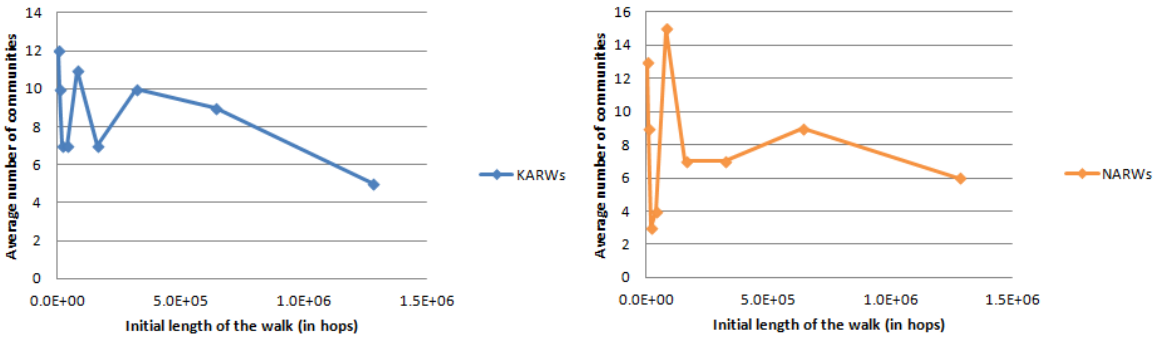


Figure 5.24: Number of communities in the main graph.

Table 5.5: Average number of communities detected by KARWs and NARWs

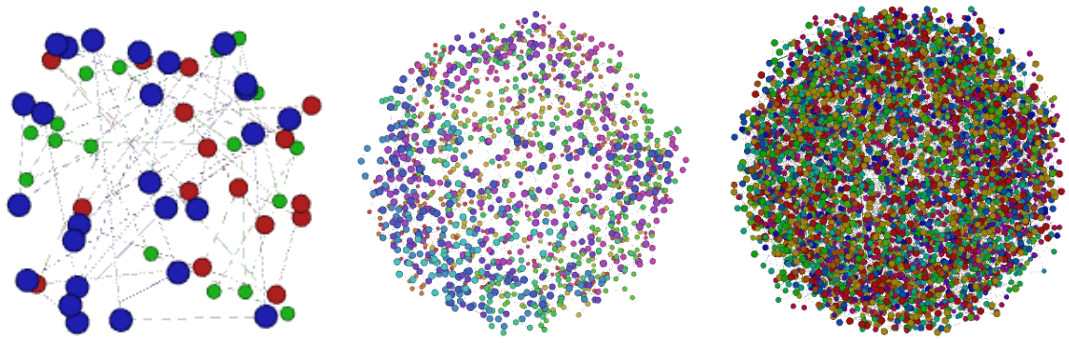
L_w $(in\ hops)$	$KARWs$	$NARWs$
5000	12	13
10000	10	9
20000	7	3
40000	7	4
80000	11	15
160000	7	7
320000	10	7
640000	9	9
1280000	5	6



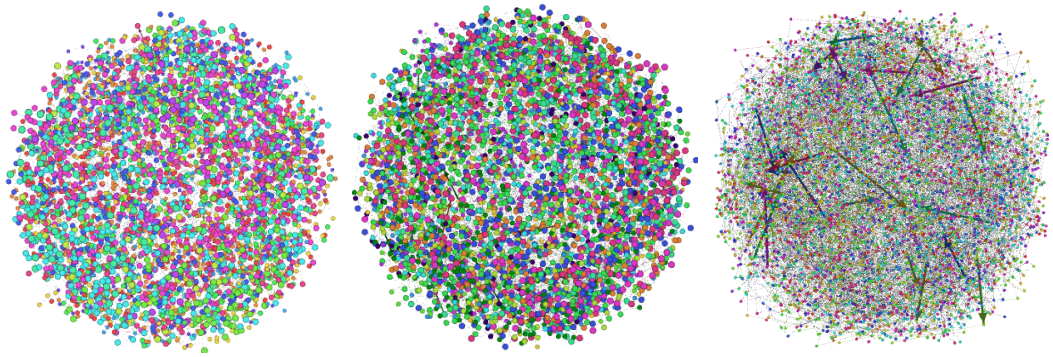
(a) KARWs (b) NARWs

Figure 5.25: Comparison of detected communities between KARWs and NARWs

It has been observed from Table 5.5 that for KARWs, the maximum number of communities detected using this algorithm is 12 when the initial length of the walk is 5 Khops. In general, the number of detected communities are fluctuating with a tendency of less number of communities being detected as the intended lengths of walk get higher, depicted in Figure 5.25a. The reason behind this is, when the intended lengths of the walk get higher, there are more nodes re-visited a number of times, means more connectivity among the nodes causing a reduction on the number of communities in the network. For NARW, the highest number of communities detected is 15 when the initial length of the walk is 80 Khops, illustrated in Table 5.5. Number of detected communities using NARW algorithm also fluctuates with the initial lengths being increased shown in Figure 5.25b. When the initial lengths of the walk reach upto a maximum of 1 Mhops, the number of communities detected using KARW and NARW are 5 and 6 respectively as shown in Figure 5.30 and Figure 5.31. In the following graphical representations, we can see that nodes from the same communities are identified with similar colors, therefore, different communities are identified in subgraphs with different colors. Bigger communities with increasing number of nodes are more visible than the rest. It has been estimated that 213 communities are being detected in the main graph.

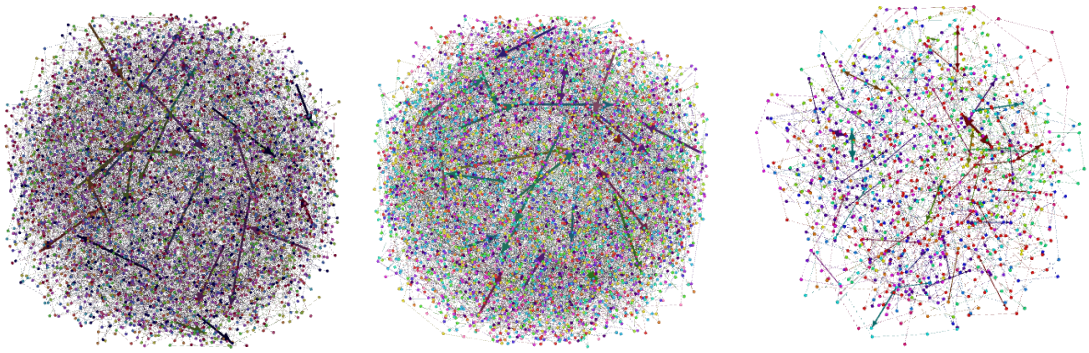


(a) K=1 (b) K=2 (c) K=4



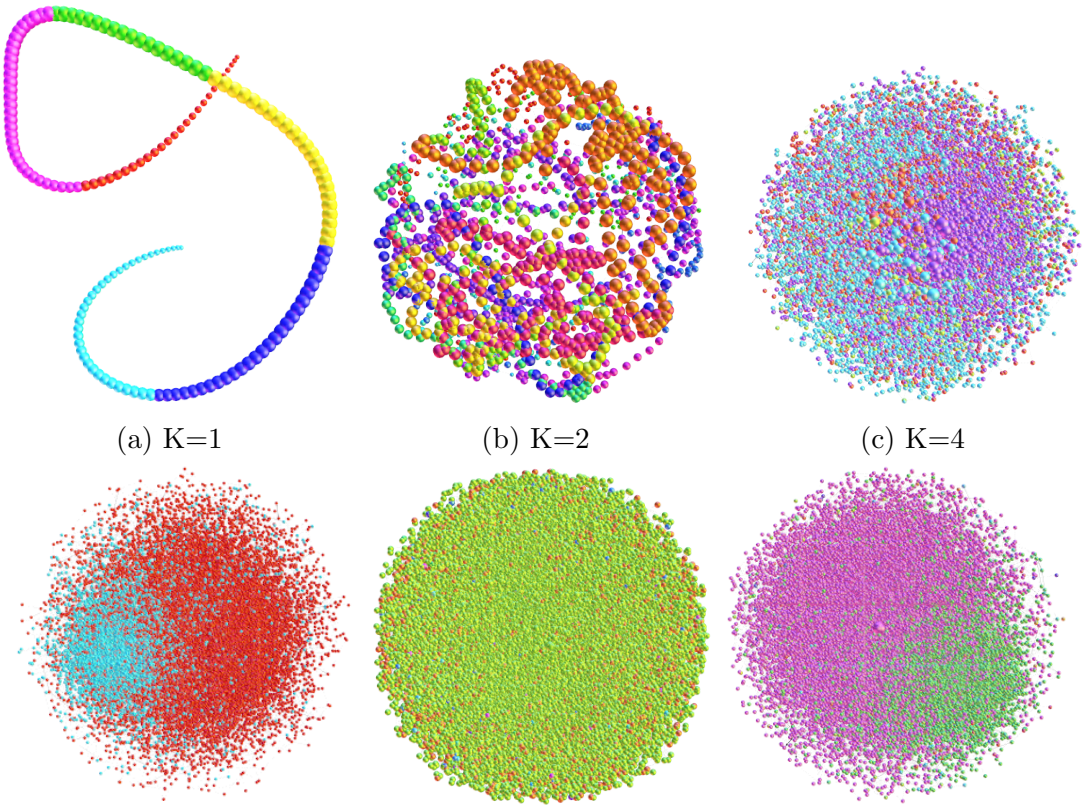
(d) K=8 (e) K=16 (f) K=32

Figure 5.26: Modularity of sampled graphs for KARWs with 5 Khops walk



(a) N=1 (b) N=2 (c) N=4

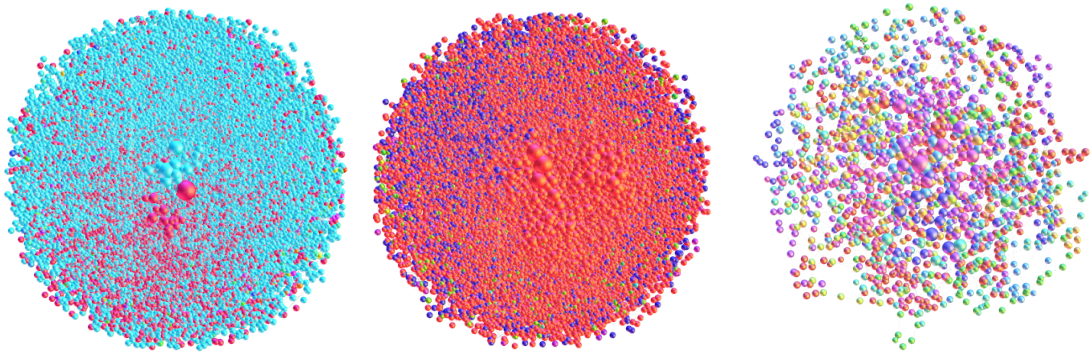
Figure 5.27: Modularity of sampled graphs for NARWs with 5 Khops walk.



(a) K=1 (b) K=2 (c) K=4

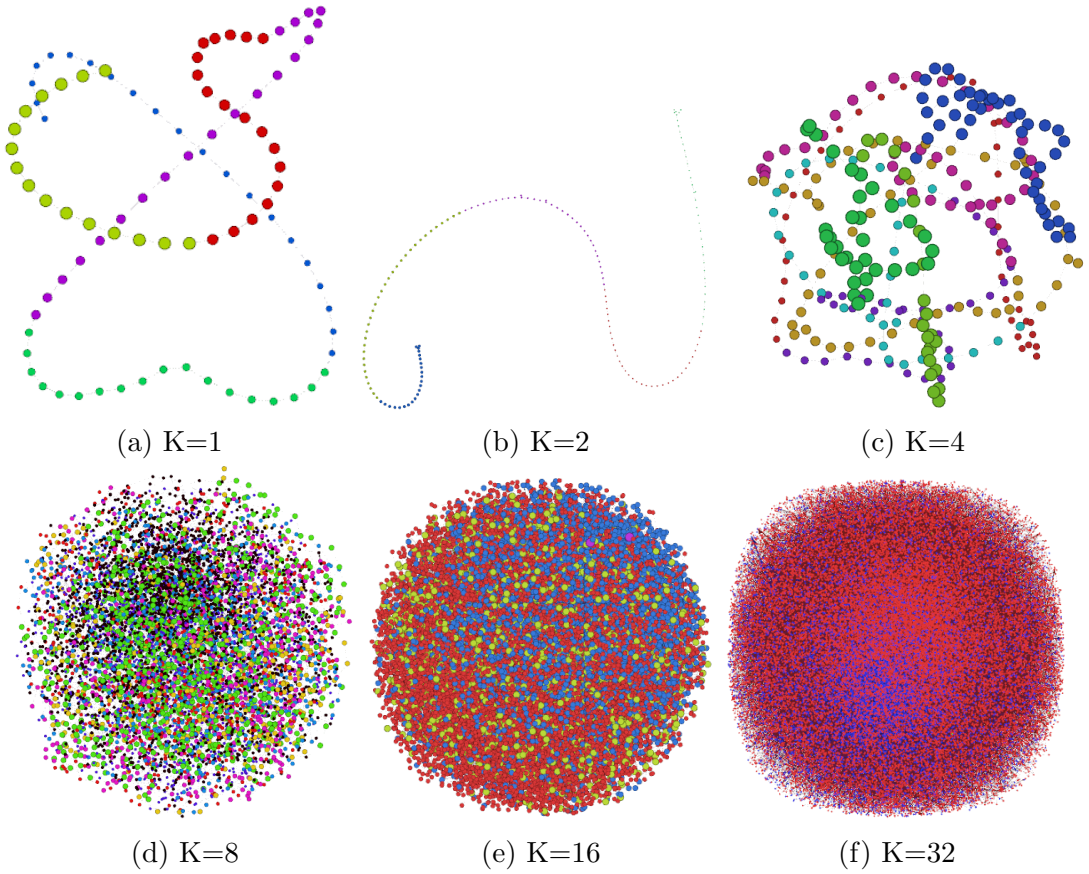
(d) K=8 (e) K=16 (f) K=32

Figure 5.28: Modularity of sampled graphs for KARWs with 80 Khops walk.



(a) N=1 (b) N=2 (c) N=4

Figure 5.29: Modularity of sampled graphs for NARWs with 80 Khops walk.



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

Figure 5.30: Modularity of sampled graphs for KARWs with 1 Mhops walk.

(a) N=1 (b) N=2 (c) N=4

Figure 5.31: Modularity of sampled graphs for NARWs with 1 Mhops walk.

5.1.8 Clustering Coefficient

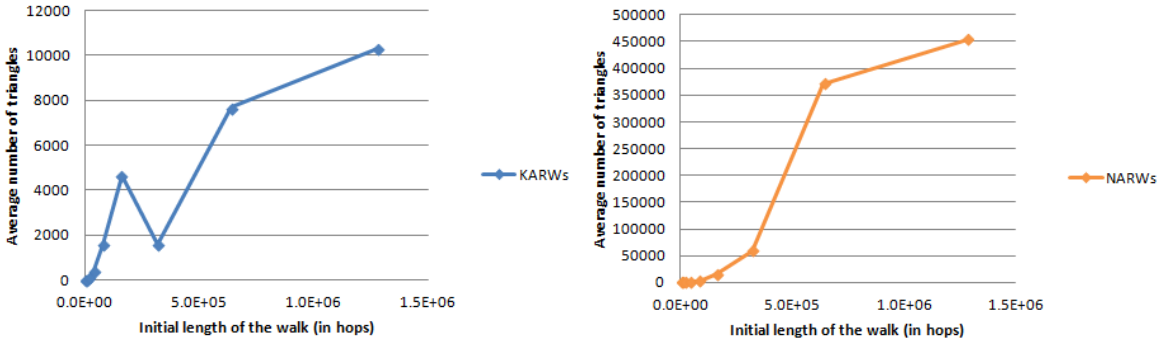
The mean value for the average clustering coefficient with the intended lengths of walk from 5 Khops to 1 Mhops are estimated to be 0.0156 for KARWs and 0.0312 for NARWs, calculated automatically by Gephi [8] using algorithm in [28]. Detail values of clustering coefficient are mentioned from Table A.7 to Table A.15. It is worthy to note here that the average clustering coefficient for $K = 1$ in KARW is zero since no cluster was formed due to restriction of re-visiting nodes. Another key feature is, average number of triangles encountered are estimated from Table A.7 to Table A.15 while calculating clustering coefficient and shown in Table 5.7. We are going to show the average number of triangles found using the two proposed algorithms and compare these to the total number of triangles in the main graph depicted in the following Figure 5.6:

Table 5.6: Average number of triangles of KARWs and NARWs with original graph.

Original Graph	$L_w = 5Khops$		$L_w = 80Khops$		$L_w = 1280Khops$	
	KARWs	NARWs	KARWs	NARWs	KARWs	NARWs
2228135	21	23	1595	2713	10298	453822

Table 5.7: Average number of triangles in KARWs and NARWs.

L_w *(in hops)*	$KARWs$	$NARWs$
5000	21	23
10000	32	52
20000	116	260
40000	436	937
80000	1595	2713
160000	4670	15674
320000	1604	59255
640000	7692	371380
1280000	10298	453822



(a) KARW (b) NARW

Figure 5.32: Comparison between KARWs and NARWs based on average number of triangles encountered

In case of KARW, the number of triangles increase with the increase of initial lengths with an exception when initial length of the walk is 320 Khops. Further investigation is required for this specific case. On the other hand, in NARW algorithm, number of triangles are proportional to initial lengths of the walk depicted in Figure 5.32b. There were no triangles formed during $K = 1$ in KARW. It also shows from Table 5.7 that NARW has more triangles encountered than KARW because NARW has more freedom of movement between nodes while sampling graphs. Clustering coefficient of the main graph has been estimated to be 0.23 and 2,228,135 triangles have been encountered from the main graph.

Chapter 6

Conclusions and Future Directions

6.1 Final Remark

Online Social Networks (OSNs) play a very important role in the 21st century and it has now become significant in every aspects of our lives. It is a very powerful media where people get influenced by others very quickly and information, news, gossip of any kind spread at a very fast rate. Therefore, much of the attentions are required to OSNs for various interests such as business, politics, health and well-being, economics and so forth. Large organizations nowadays are choosing most influential people as their potential customers to promote their products using OSNs. New products are also being developed and introduced to customers with the help of OSNs platform. Therefore, effective and efficient sampling techniques are necessary in order to find potential customers online to promote and to improve business. Only the right way of gathering and overlaying data would help any organization to decide which potential customers would be the most valuable ones to approach. It is worth to remember that ubiquitous social networks like Facebook and Twitter are ten years and seven years old, respectively and we do not know what new technologies lie ahead. It will be a challenge for the business people to understand how strategies are evolving in existing social media and adapt newer techniques in the years to come.

With this research, we are able to find out how sampling can be done in most effective and efficient way on a large social graph by introducing the two newly proposed random walk-based techniques called K-Avoiding Random Walk (KARW) and Neighborhood-Avoiding Random Walk (NARW) in particular. Then we setup the parameters and apply these algorithms alongside with three existing state-of-the-art

algorithms to the Facebook dataset. Afterwards, we calculated simulation results of KARWs and NARWs and compared with the existing algorithms based on the performance metrics and then observed sampled subgraphs based on some of the key statistical features and tried to compare with the original graph. Based on the metrics, both of our algorithms outperformed existing state-of-the-art algorithms based on sampling unique nodes. Subgraphs constructed from sampling nodes from the original Facebook graph were analyzed with key statistical features such as degree distribution, centrality measurements, modularity and clustering coefficient and found out that the subgraphs inherited all the properties of the original graph as was expected at the beginning of our research. If we summarize our research work, we can say that KARW was able to minimize the number of re-visits by a controlling parameter, K, while walking on the graph and NARW was able to push the sampling technique to different region of the graph deliberately.

Besides being good sampling techniques, they both suffer in some of the areas that we observed throughout simulation and analysis phases. For KARWs, the intended lengths of the walk were not achieved even though the restrictions were more relaxed (i.e., when $K = 32$ in KARW). The walks also terminated prematurely when restrictions were at its best ($K = 1$ and K =2) causing small number of nodes being sampled. With NARWs, the growth of the neighborhood size was exponential and the algorithm terminated prematurely, therefore, it was not possible to sample substantial amount of nodes when $N = 4$ in NARW.

6.2 Future Directions

We would also like to highlight some of the clues and ideas for researchers into their future endeavor. They are summarized as follows:

- A random walk is a stochastic process that starts at one node of a graph, and moves from the current node to an adjacent node, at each step, chosen randomly from the neighbors of the current node. Graph exploration problems such as hunting or tracking on a graph are particularly interesting where the environments are unknown and in that case multiple random walks can be used to traverse the graph and calculate the time to cover the graph, which is an important measure of the efficiency of random walks. This was not the intention of our research work but we can leave this strategy as the future endeavor.

- The main purpose of this research is to provide a sampling tool that can be easily adjusted with different parameter settings while selecting the value of K and N in KARW and NARW respectively depends on statistical features of the graph under investigation. The relationship between key statistical features of graphs and choosing the correct values of K and N is complicated and thus subject to the goal that we can to achieve with sampling. We leave this as an interesting future research.

- We can think about a hybrid approach in future where it may be possible to amalgamate the characteristics of both K-Avoiding and N-Avoiding random walks to develop a new technique for sampling nodes which would may perform better than many of the existing strategies.

- It would be more effective to see other random walk strategies besides what we have discussed in this research, that would help us to get some new ideas and directions near future.

- Another direction would be a random walk starting from the same source and observe the patterns of the walks rather than starting random walk choosing purely randomly.

- Lengths of the intended walk would also have to be extended over to few million hops and observe the nature of patterns of the walks.

- These proposed algorithms can also be applied to some other datasets (such as Twitter, YouTube, QQ and so forth) besides being used dataset only from Facebook and analyze, compare social behavior of these complex networks.

Appendix A

Additional Information

A.1 Length of the walk completed by KARWs and NARWs

Table A.1: Length of the walk completed by KARWs and NARWs

L_w (in hops)	KARW1	KARW2	KARW4	KARW8	KARW16	KARW32
5K	34	619	5000	5000	5000	5000
10K	36	2126	7013	10000	10000	10000
20K	143	985	7027	14057	20000	20000
40K	121	468	7227	7013	10819	12207
80K	288	1237	5187	20748	35697	55482
160K	165	1169	4933	37832	49072	87429
320K	179	907	7370	51639	60872	89391
640K	153	1125	7899	48392	63849	85744
1280K	220	810	4286	509348	64572	91369

Table A.2: Length of the walk completed by NARWs

L_w (in hops)	NARW1	NARW2	NARW4
5K	5000	5000	5000
10K	10000	10000	335
20K	20000	20000	2557
40K	40000	40000	3162
80K	80000	80000	9792
160K	160000	160000	2430
320K	320000	305434	1657
640K	640000	453766	6297
1280K	819200	904194	12665

A.2 Nodes sampled over entire population

Table A.3: KARW vs (URS, RW, MHRW) by nodes sampled nodes over entire population (%)

L_w (in hops)	URS	RW	MHRW	KARW1	KARW2	KARW4	KARW8	KARW16	KARW32
5K	8	5.73	3.1	.054	0.862	6.73	6.7	7.8	8.26
10K	16	12	11.65	0.057	7.17	12.24	9.04	12.11	9.03
20K	27.43	20	11	0.7	1	9	2.1	15.97	16.37
40K	47.34	31.23	20	0.3	1	8	9.03	8.7	13.28
80K	72.65	44.17	34	0.3	2	6	5.28	11.15	21.46
160K	92	58.35	52	0.1	2	6	22.58	4.68	13.27
320K	99	70.45	70	0.5	1	9	10.76	14.03	12.87
640K	100	81.32	82	0.02	2	9	9.04	12.19	5.3
1280K	100	89.45	90	0.01	1	6	6.75	5.26	3.09

Table A.4: NARWs vs (URS, RW, MHRW) by sampled nodes over entire population (%)

L_w (in hops)	URS	RW	MHRW	NARW1	NARW2	NARW4
5K	8	5.73	3.1	7.9	7.65	6.87
10K	16	12	11.65	11.97	11.98	2.41
20K	27.43	20	11	20	19.89	2.85
40K	47.34	31.23	20	31.74	30.68	2.73
80K	72.65	44.17	34	43.54	43.57	8.77
160K	92	58.35	52	58.3	56.39	2.67
320K	99	70.45	70	70.12	67.84	1.95
640K	100	81.32	82	80.43	72.59	4.78
1280K	100	89.45	90	88.75	80.675	6.74

A.3 Nodes sampled exactly once

Table A.5: KARWS vs (URS, RW, MHRW) by nodes sampled exactly once (%)

L_w (in hops)	URS	RW	MHRW	KARW1	KARW2	KARW4	KARW8	KARW16	KARW32
20K	85	66	52	100	92	80	72	66	73
40K	72	52	46	100	94	74	81	76	75
80K	50	40	36	100	91	77	93	65	94
160K	22	29	22	100	93	82	76	64	96
320K	3	21	9	100	93	78	77	75	87
640K	0	15	3	100	89	74	76	88	93
1280K	0	10	1	100	93	85	85	77	82

Table A.6: NARWs vs (URS, RW, MHRW) by nodes sampled exactly once (%)

L_w (in hops)	URS	RW	MHRW	NARW1	NARW2	NARW4
20K	85	66	52	66	65	74
40K	72	52	46	52	52	68
80K	50	40	36	40	40	61
160K	22	29	22	29	29	73
320K	3	21	9	21	22	74
640K	0	15	3	15	18	56
1280K	0	10	1	10	13	46

A.4 Statistical data of sampled graphs for KARWs and NARWs

Table A.7: Statistical data of sampled graph when K=1 in KARW

L_w	AvgDeg	Diameter	APL	#Short.Path	Modularity	#Communities	ClustCoeff.	#Triangles
5000	0.983	59	20.33	1770	0.682	3	0	0
10000	0.972	35	12.33	630	0.589	3	0	0
20000	0.75	3	1.66	6	0	1	0	0
40000	0.997	309	103.66	47895	0.863	9	0	0
80000	0.994	162	54.66	13203	0.795	6	0	0
160000	0.994	158	53.33	12561	0.763	5	0	0
320000	0.997	359	120.33	64620	0.873	10	0	0
640000	0.997	356	119.33	63546	0.874	10	0	0
1280000	0.989	92	31.33	4278	0.704	4	0	0

Table A.8: Statistical data of sampled graph when K=2 in KARW

L_w	AvgDeg	Diameter	APL	#Short.Path	Modularity	#Communities	ClustCoeff.	#Triangles
5000	1.083	228	78.36	1910528	0.89	14	0.007	5
10000	1.088	188	63.13	1431617	0.861	11	0.01	7
20000	1.102	199	64.2688	4052595	0.884	15	0.013	13
40000	1.085	184	73.4225	2109903	0.884	13	0.009	8
80000	1.086	182	64.5712	2288141	0.883	15	0.009	7
160000	1.16	22	7.8786	305	0.451	2	0	0
320000	1.092	205	71.1582	2965302	0.879	13	0.01	9
640000	1.078	181	68.4906	1320291	0.883	14	0.004	3
1280000	1.025	154	52.7269	12890	0.79	6	0	0

Table A.9: Statistical data of sampled graph when K=4 in KARW

L_w	AvgDeg	Diameter	APL	#Short.Path	Modularity	#Communities	ClustCoeff.	#Triangles
5000	1.18	107	34.98	17930994	0.83	14	0.012	36
10000	1.181	91	31.543	15566980	0.83	14	0.014	31
20000	1.11	148	51.1926	4565082	0.879	15	0.008	11
40000	1.13	151	44.1504	6612948	0.869	16	0.012	17
80000	1.42	63	18.4456	127136900	0.601	4	0.015	151
160000	1.308	85	21.9937	58369605	0.661	7	0.011	69
320000	1.273	83	23.8716	48149721	0.702	8	0.015	72
640000	1.161	112	37.7553	19202066	0.84	13	0.014	34
1280000	1.07	163	61.9381	51470	0.825	7	0.013	2

Table A.10: Statistical data of sampled graph when K=8 in KARW

L_w	AvgDeg	Diameter	APL	#Short.Path	Modularity	#Communities	ClustCoeff.	#Triangles
5000	1.176	99	33.493	17829531	0.835	13	0.012	29
10000	1.174	118	34.4819	14428602	0.845	15	0.012	27
20000	1.552	47	14.814	162728302	0.419	2	0.018	242
40000	1.931	44	11.6687	414998016	0.39	2	0.023	815
80000	2.107	39	10.7649	507263007	0.363	2	0.025	1188
160000	1.644	47	13.8979	239769750	0.476	3	0.018	332
320000	1.463	64	16.7297	125406402	0.58	5	0.018	186
640000	2.131	39	10.6738	522671044	0.388	2	0.026	1310
1280000	1.333	65	20.2721	52823831	0.648	6	0.015	82

Table A.11: Statistical data of sampled graph when K=16 in KARW

L_w	AvgDeg	Diameter	APL	#Short.Path	Modularity	#Communities	ClustCoeff.	#Triangles
5000	1.182	99	33.493	17829531	0.831	14	0.012	29
10000	1.292	75	22.5553	58675600	0.69	8	0.012	64
20000	1.417	70	17.6793	98366852	0.621	5	0.017	168
40000	1.98	46	11.1845	395075268	0.395	2	0.024	883
80000	2.737	32	8.8233	807525889	0.104	19	0.035	4125
160000	2.836	33	8.6405	851238976	0.446	19	0.035	4663
320000	2.807	30	8.64	853457796	0.417	11	0.037	4733
640000	2.946	31	8.43	896942601	0.41	8	0.038	5555
1280000	3.496	27	7.7534	1151991481	0.471	3	0.047	11489

Table A.12: Statistical data of sampled graph when K=32 in KARW

L_w	AvgDeg	Diameter	APL	#Short.Path	Modularity	#Communities	ClustCoeff.	#Triangles
5000	1.176	99	33.493	17829531	0.832	14	0.012	29
10000	1.292	75	22.5553	58675600	0.712	9	0.012	64
20000	1.549	60	14.9801	161747537	0.468	4	0.018	259
40000	1.973	44	11.1241	397384299	0.375	2	0.024	895
80000	2.751	30	8.5898	798853696	0.409	20	0.033	4099
160000	4.134	24	7.106	1347440556	0.464	8	0.057	22956
320000	2.827	28	8.4986	862303225	0.434	14	0.033	4625
640000	4.737	25	6.7453	1577996176	0.399	5	0.065	39248
1280000	5.1	24	6.5577	1659340225	0.387	2	0.069	50212

Table A.13: Statistical data of sampled graph when N=1 in NARW

L_w	AvgDeg	Diameter	APL	#Short.Path	Modularity	#Communities	ClustCoeff.	#Triangles
5000	1.176	99	33.4924	17821098	0.833	14	0.012	29
10000	1.293	76	22.5303	58453719	0.792	9	0.012	65
20000	1.549	60	14.9794	161747537	0.514	4	0.018	259
40000	1.973	44	11.1231	397384294	0.403	2	0.024	895
80000	2.749	33	8.5626	799334256	0.652	16	0.034	3990
160000	4.11	26	7.00341	1362163556	0.395	6	0.055	22663
320000	6.489	19	5.8789	1987420980	0.444	5	0.089	121220
640000	10.271	21	6.3669	2147483647	0.379	5	0.139	555669
1280000	5.1	24	6.5577	1659340225	0.378	5	0.181	1291629

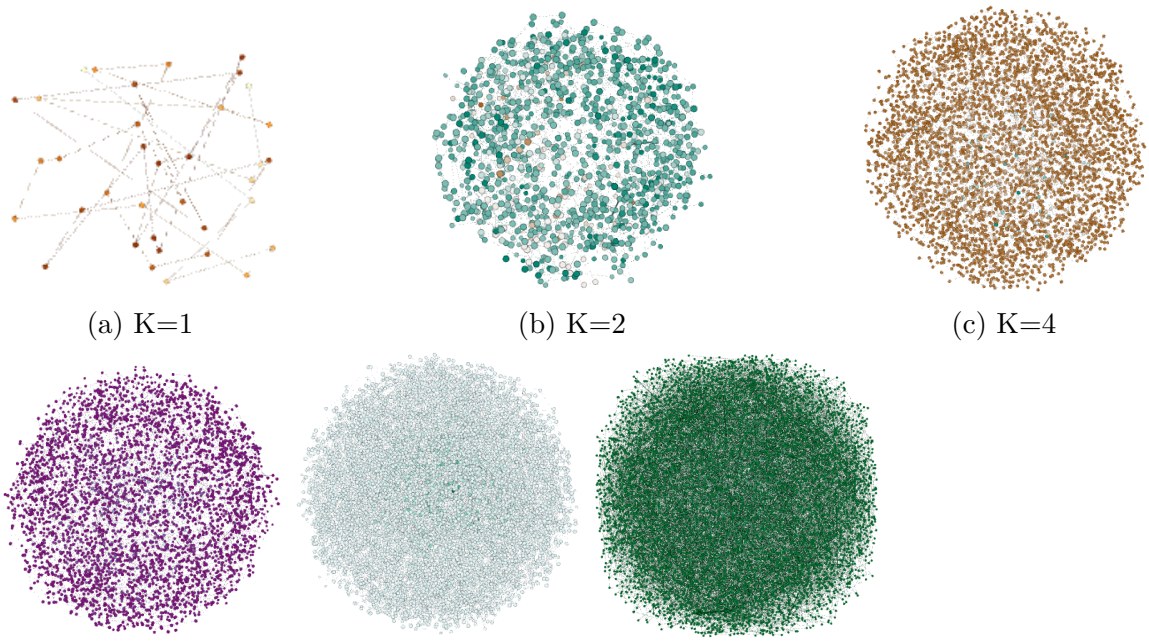
Table A.14: Statistical data of sampled graph when N=2 in NARW

L_w	AvgDeg	Diameter	APL	#Short.Path	Modularity	#Communities	ClustCoeff.	#Triangles
5000	1.175	99	33.9477	17901364	0.836	14	0.011	28
10000	1.317	87	22.0324	56205021	0.659	9	0.016	82
20000	1.549	60	14.9796	161760258	0.447	3	0.018	259
40000	1.973	44	11.1221	397364358	0.401	2	0.024	897
80000	2.735	32	8.4824	803070582	0.218	19	0.036	4130
160000	4.183	26	7.07918	1311707306	0.4	8	0.054	24334
320000	5.286	21	6.2149	1528458120	0.412	12	0.073	56544
640000	1.6	21	6.31812	2147483647	0.391	4	0.138	558084
1280000	5.552	22	6.0789	1677271070	0.155	3	0.08	69828

Table A.15: Statistical data of sampled graph when N=4 in NARW

L_w	AvgDeg	Diameter	APL	#Short.Path	Modularity	#Communities	ClustCoeff.	#Triangles
5000	1.291	93	32.8797	1123760	0.797	10	0.011	13
10000	1.357	114	40.5983	123634	0.764	8	0.029	8
20000	1.551	60	14.9358	161378912	0.442	3	0.018	262
40000	1.993	39	10.8966	385336902	0.272	9	0.027	1020
80000	1.333	68	25.5357	2464902	0.774	9	0.018	18
160000	1.3	121	38.8638	267843	0.768	8	0.021	24
320000	1.338	95	25.6802	93031	0.619	5	0.01	2
640000	1.908	37	10.2897	67757593	0.275	18	0.018	386
1280000	1.314	80	26.1218	1257813	0.78	9	0.014	10

A.5 Betweenness centrality of KARWs and NARWs with 10 Khops walk



(a) K=1 (b) K=2 (c) K=4

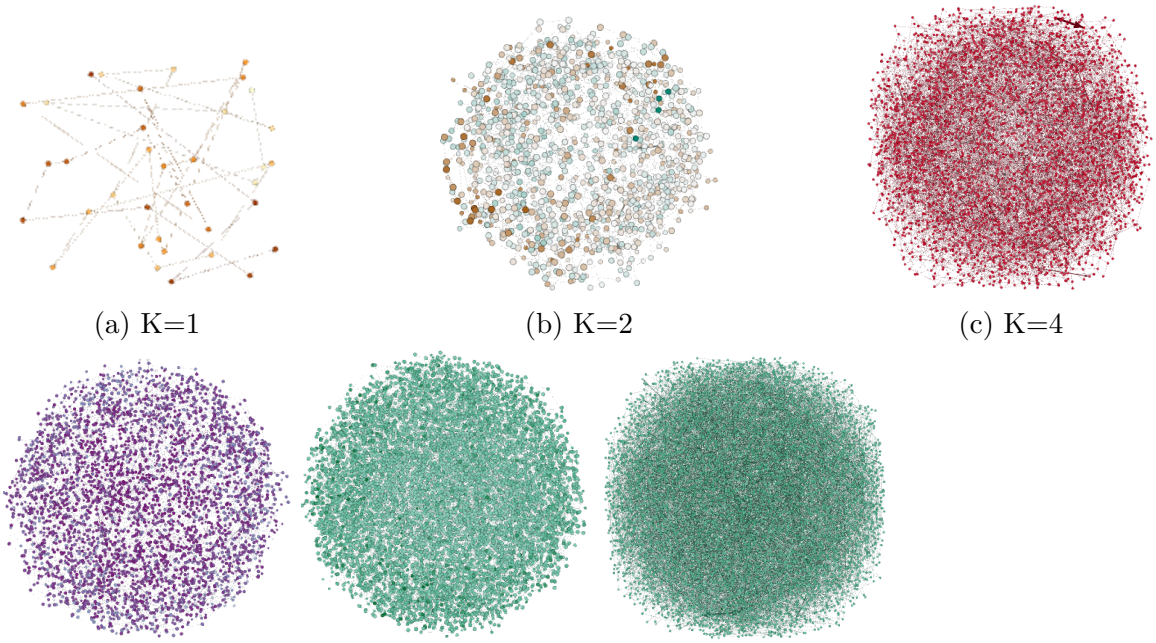
(d) K=8 (e) K=16 (f) K=32

Figure A.1: Betweenness Centrality of sampled graphs for KARWs with 10 Khops walk

(a) N=1 (b) N=2 (c) N=4

Figure A.2: Betweenness Centrality of sampled graphs for NARWs with 10 Khops walk

A.6 Closeness centrality of KARWs and NARWs with 10 Khops walk



(a) K=1 (b) K=2 (c) K=4

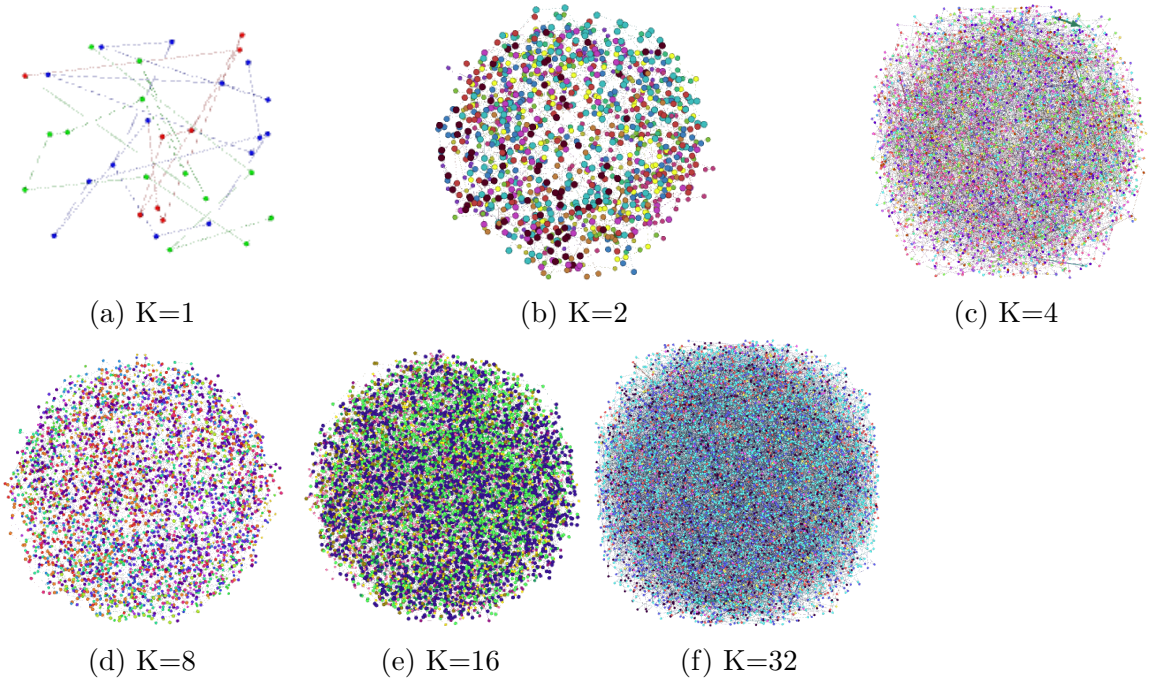
(d) K=8 (e) K=16 (f) K=32

Figure A.3: Closeness Centrality of sampled graphs for KARWs with 10 Khops walk

(a) N=1 (b) N=2 (c) N=4

Figure A.4: Closeness Centrality of sampled graphs for NARWs with 10 Khops walk

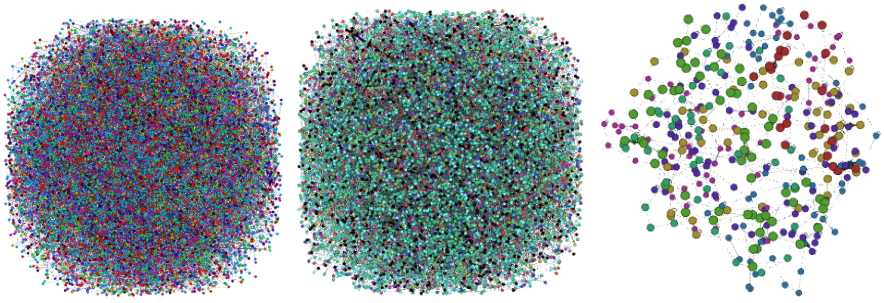
A.7 Modularity of KARWs and NARWs with 10Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

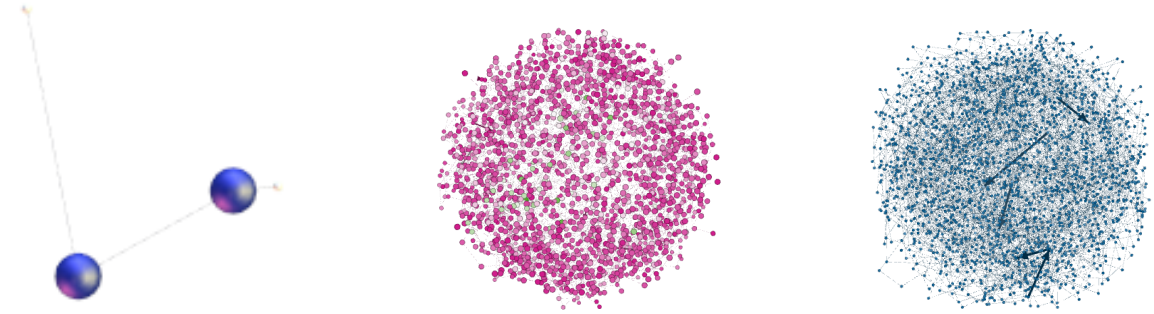
Figure A.5: Modularity of sampled graphs for KARWs with 10 Khops walk



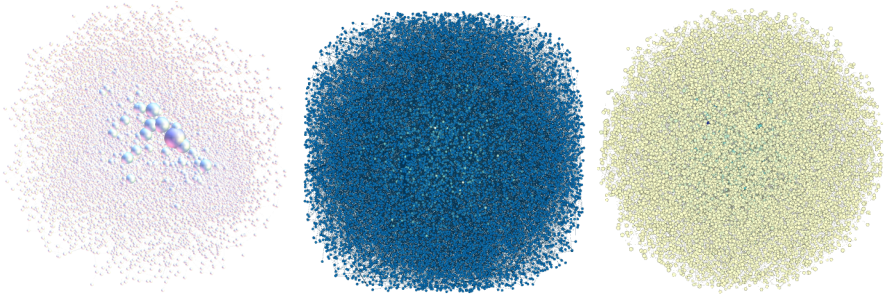
(a) N=1 (b) N=2 (c) N=4

Figure A.6: Modularity of sampled graphs for NARWs with 10 Khops walk

A.8 Betweenness centrality for KARWs and NARWs with 20 Khops walk

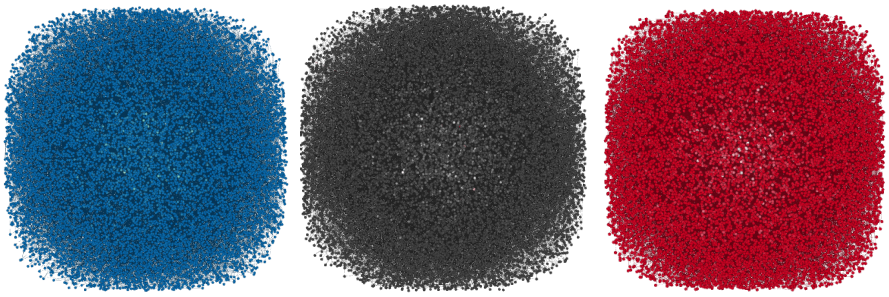


(a) K=1 (b) K=2 (c) K=4



(d) K=8 (e) K=16 (f) K=32

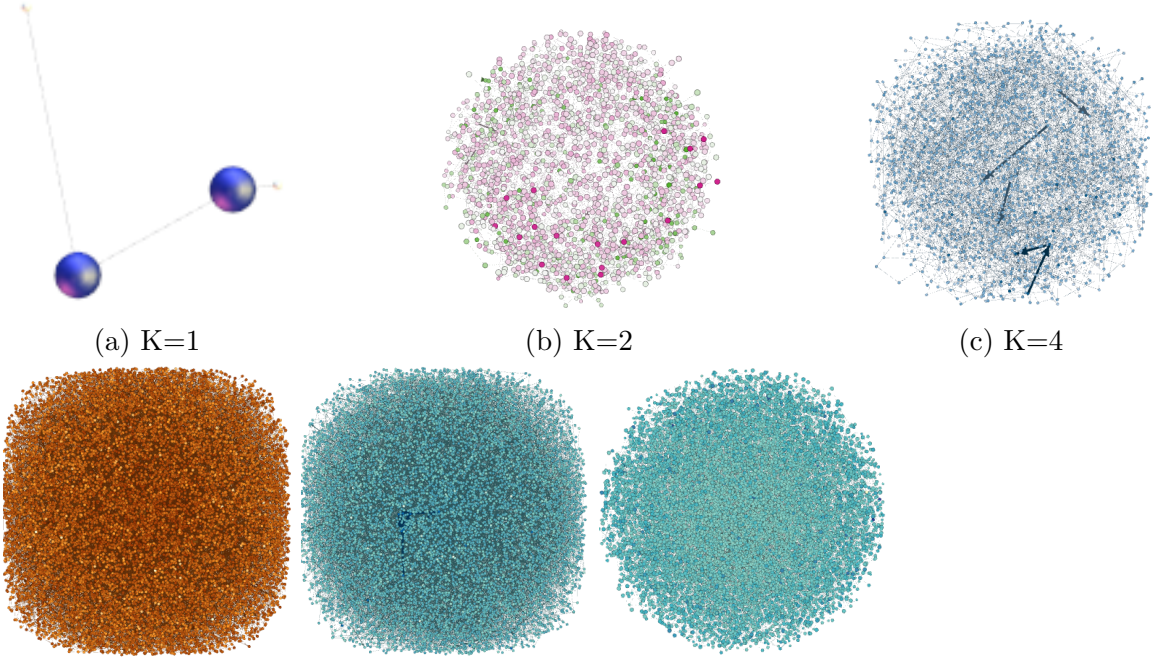
Figure A.7: Betweenness Centrality of sampled graphs for KARWs with 20 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.8: Betweenness Centrality of sampled graphs for NARWs with 20 Khops walk

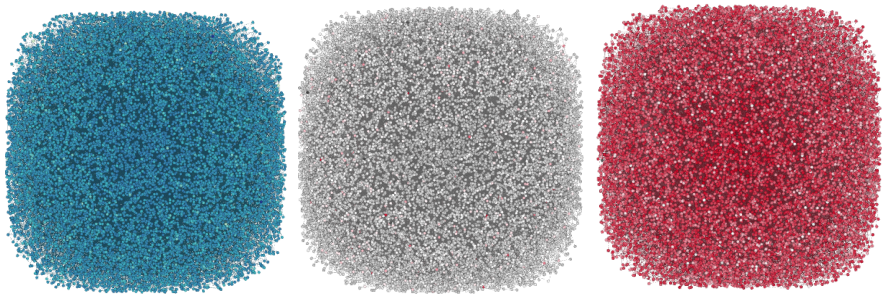
A.9 Closeness centrality of KARWs and NARWs with 20 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

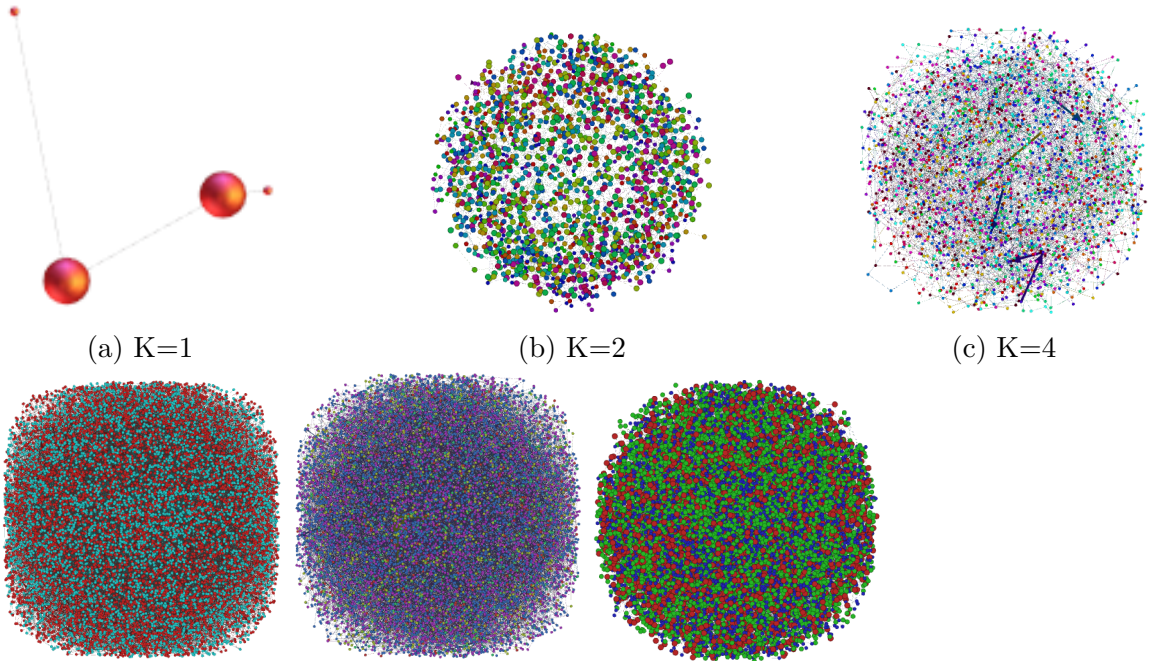
Figure A.9: Closeness Centrality of sampled graphs for KARWs with 20 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.10: Closeness Centrality of sampled graphs for NARWs with 20 Khops walk

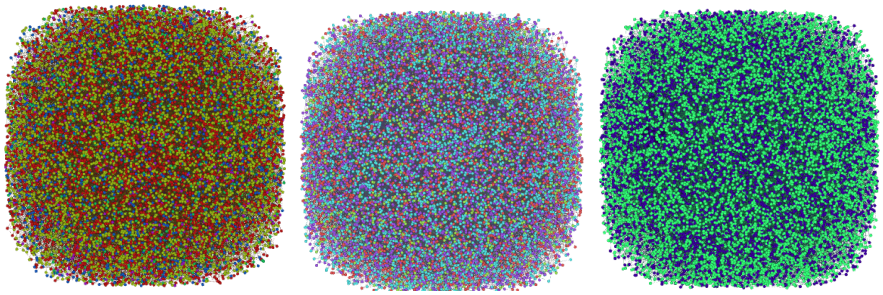
A.10 Modularity of KARWs and NARWs with 20 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

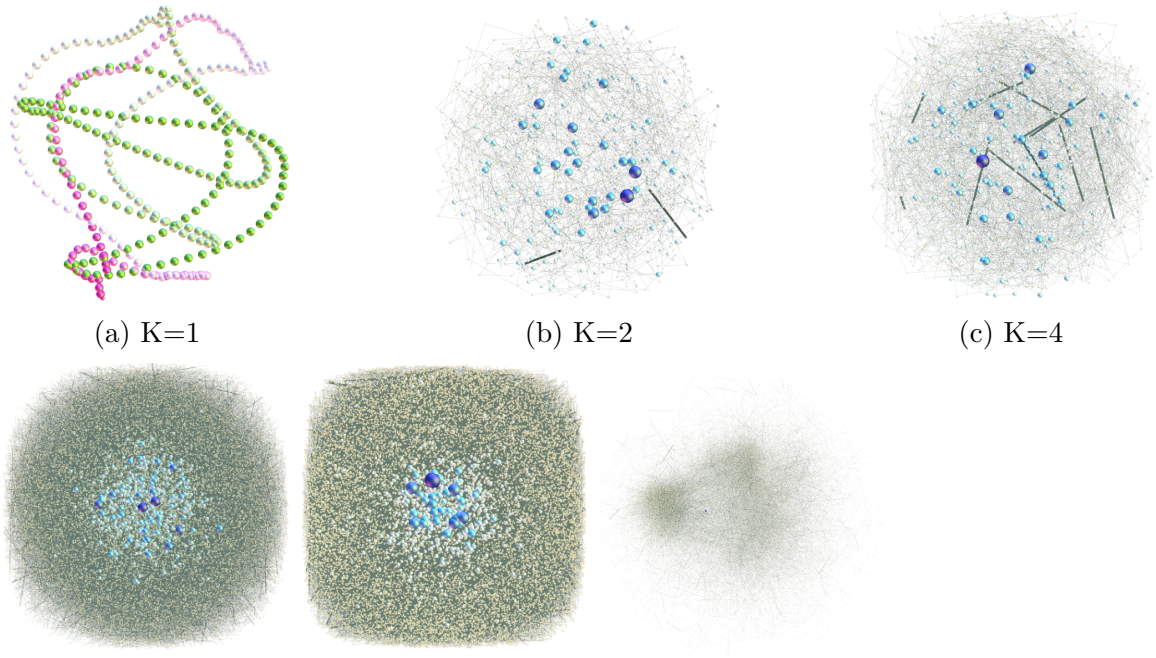
Figure A.11: Modularity of sampled graphs for KARWs with 20 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.12: Modularity of sampled graphs for NARWs with 20 Khops walk

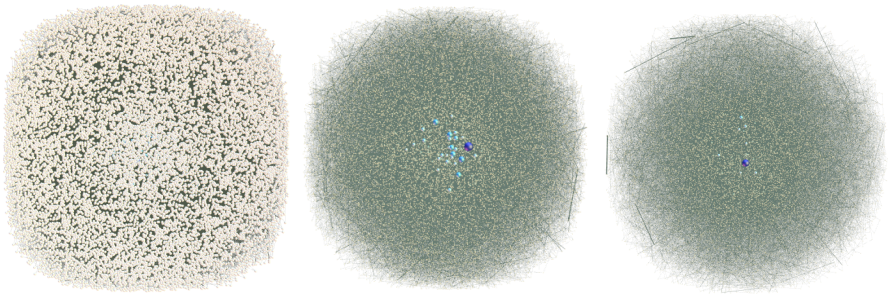
A.11 Betweenness centrality of KARWs and NARWs with 40 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

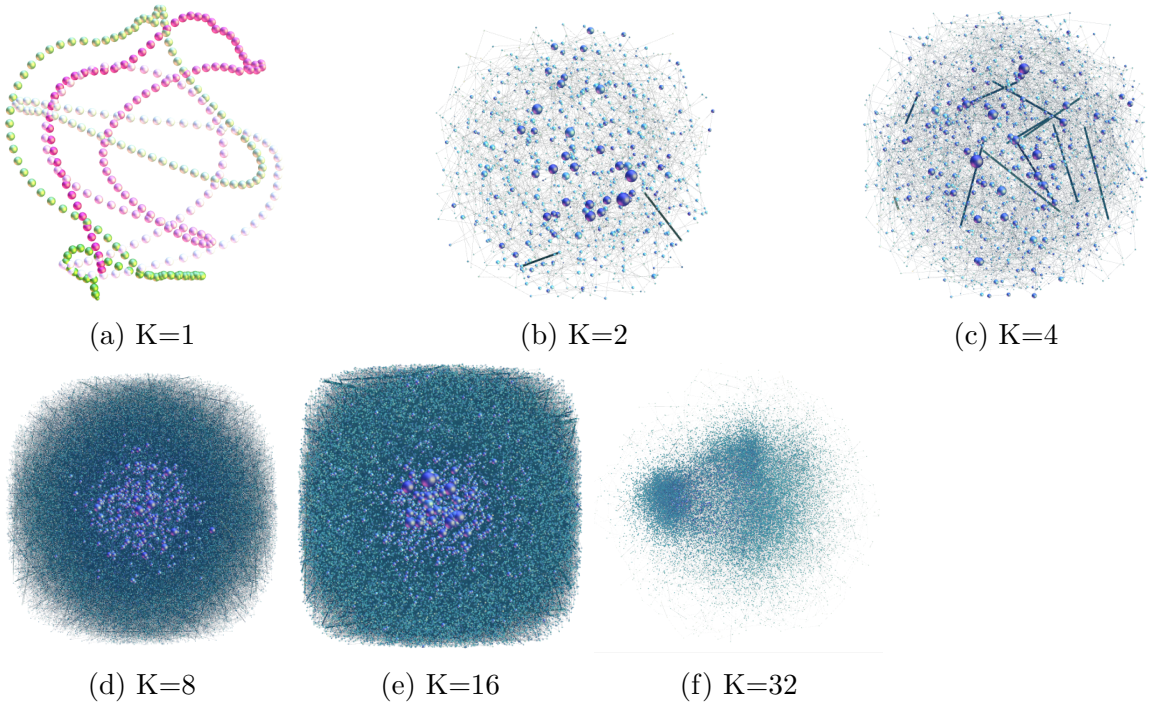
Figure A.13: Betweenness Centrality of sampled graphs for KARWs with 40 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.14: Betweenness Centrality of sampled graphs for NARWs with 40 Khops walk

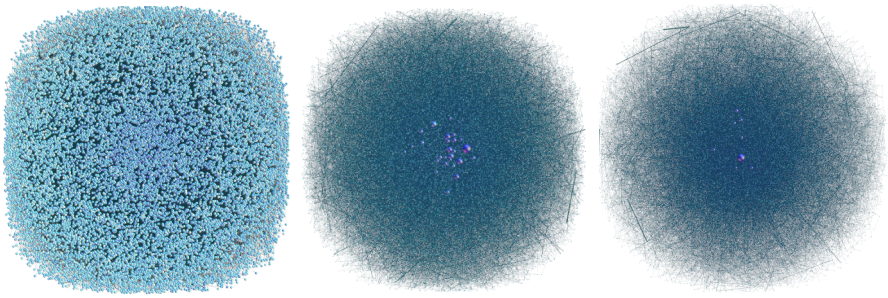
A.12 Closeness centrality of KARWs and NARWs with 40 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

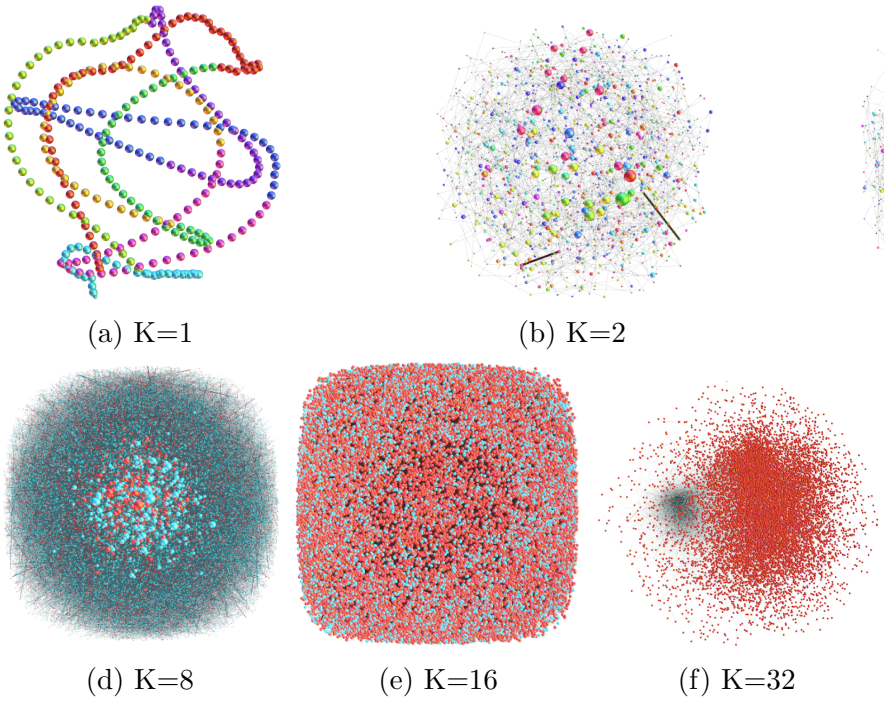
Figure A.15: Closeness Centrality of sampled graphs for KARWs with 40 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.16: Closeness Centrality of sampled graphs for NARWs with 40 Khops walk

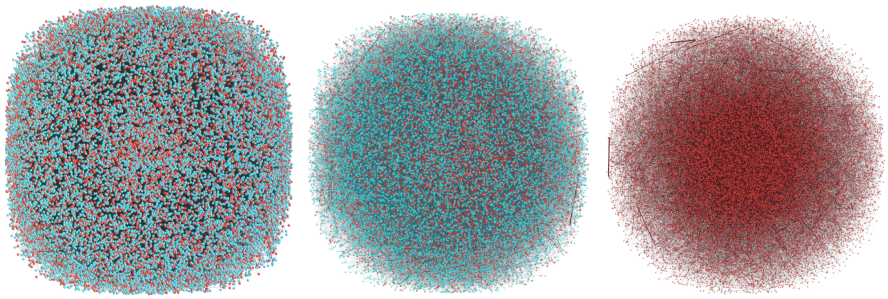
A.13 Modularity of KARWs and NARWs with 40 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

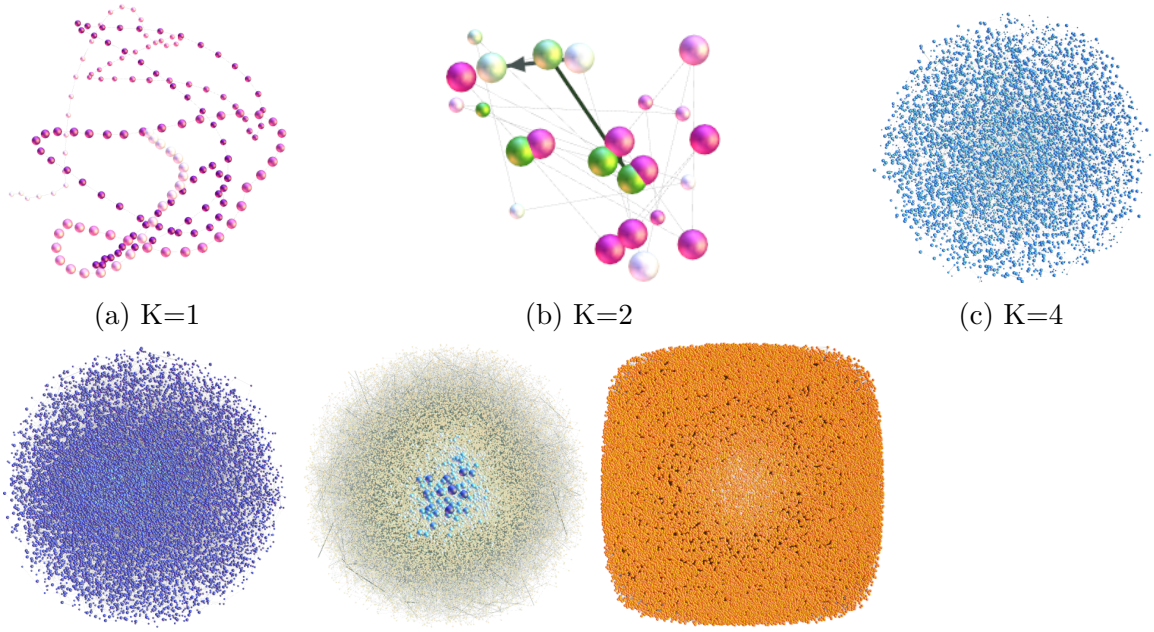
Figure A.17: Modularity of sampled graphs for KARWs with 40 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.18: Modularity of sampled graphs for NARWs with 40 Khops walk

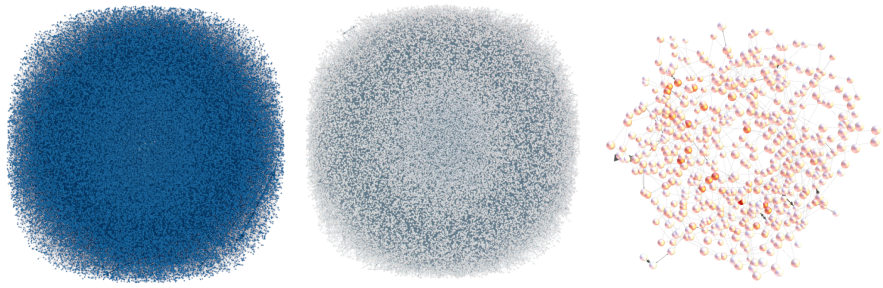
A.14 Betweenness centrality of KARWs and NARWs with 160 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

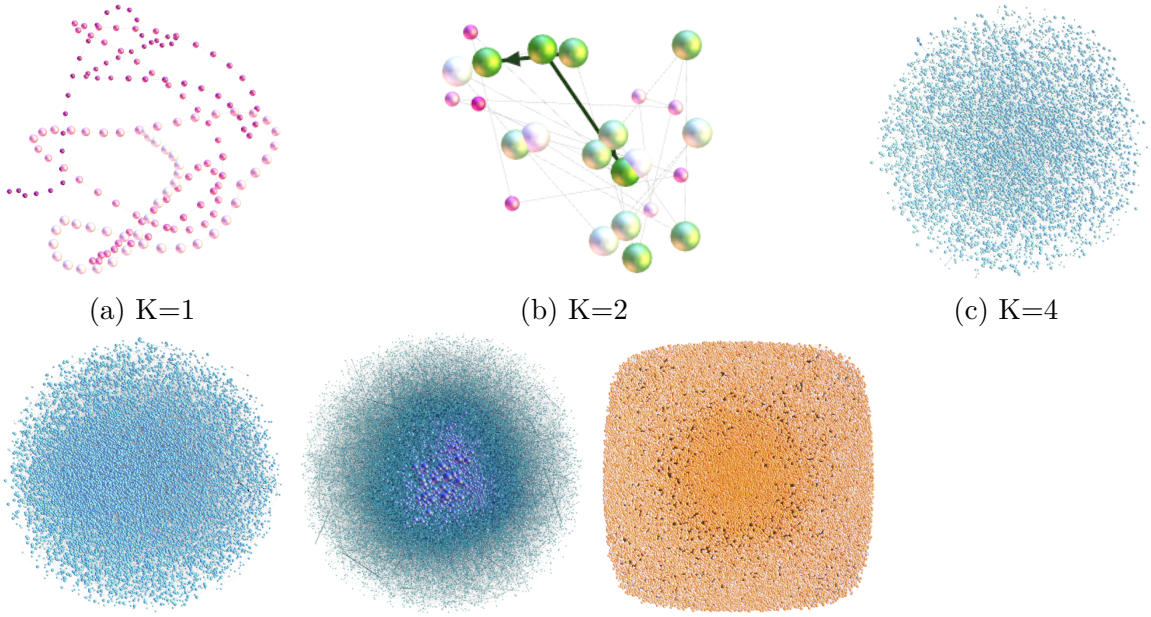
Figure A.19: Betweenness Centrality of sampled graphs for KARWs with 160 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.20: Betweenness Centrality of sampled graphs for NARWs with 160 Khops walk

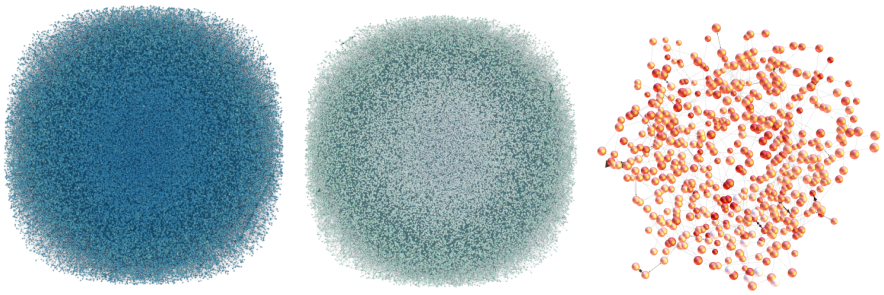
A.15 Closeness centrality of KARWs and NARWs with 160 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

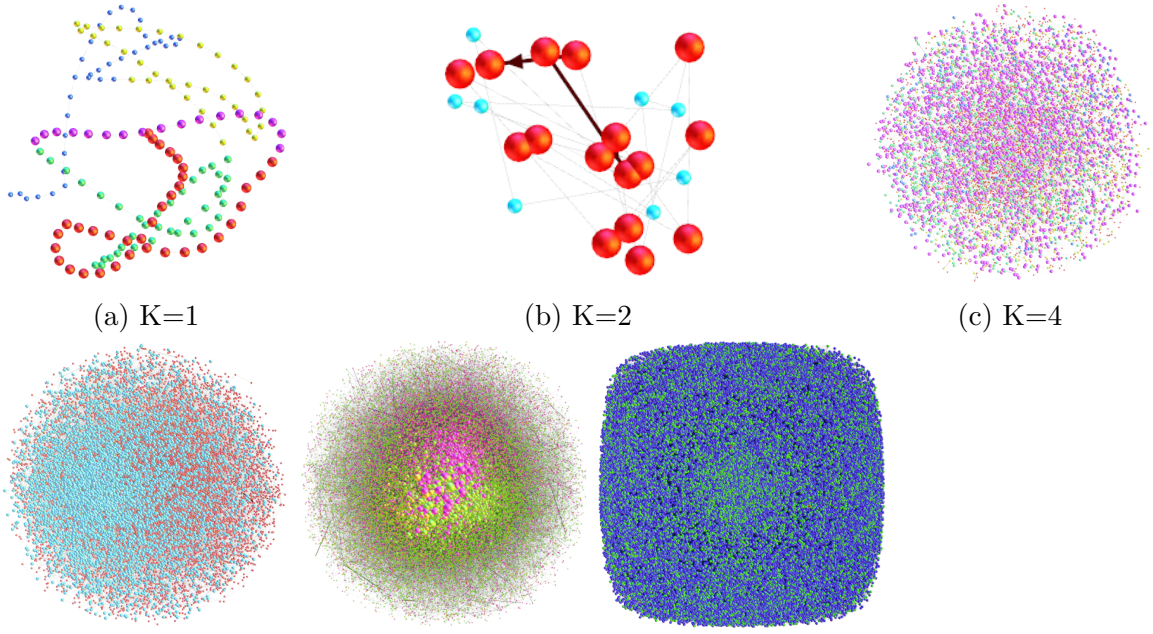
Figure A.21: Closeness Centrality of sampled graphs for KARWs with 160 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.22: Closeness Centrality of sampled graphs for NARWs with 160 Khops walk

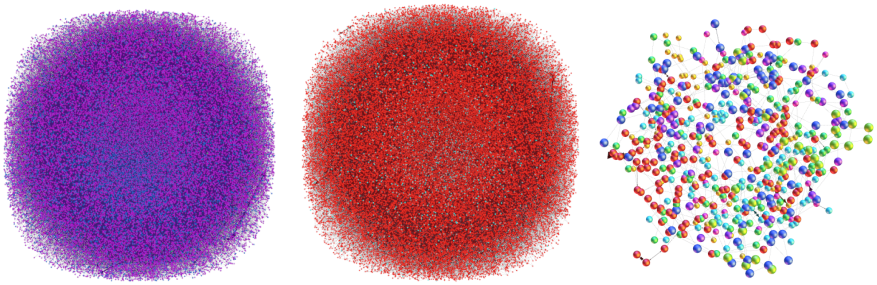
A.16 Modularity of KARWs and NARWs with 160 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

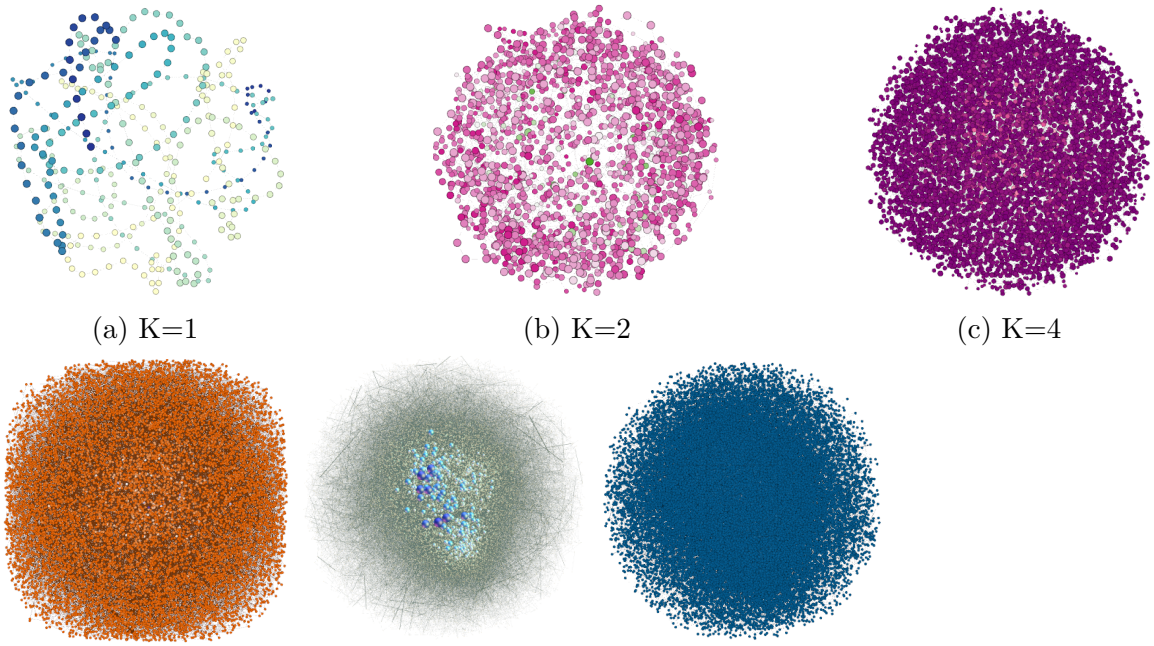
Figure A.23: Modularity of sampled graphs for KARWs with 160 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.24: Modularity of sampled graphs for NARWs with 160 Khops walk

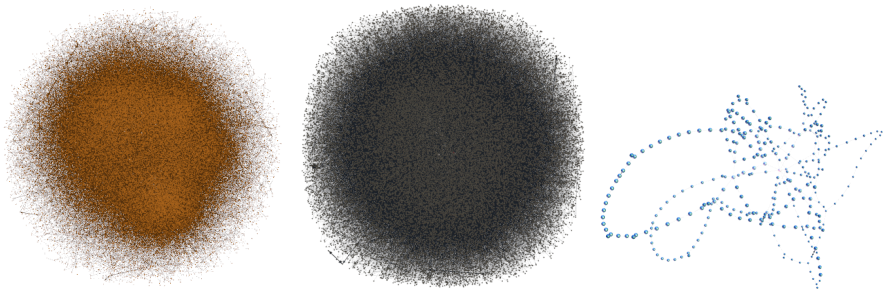
A.17 Betweenness centrality of KARWs and NARWs with 320 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

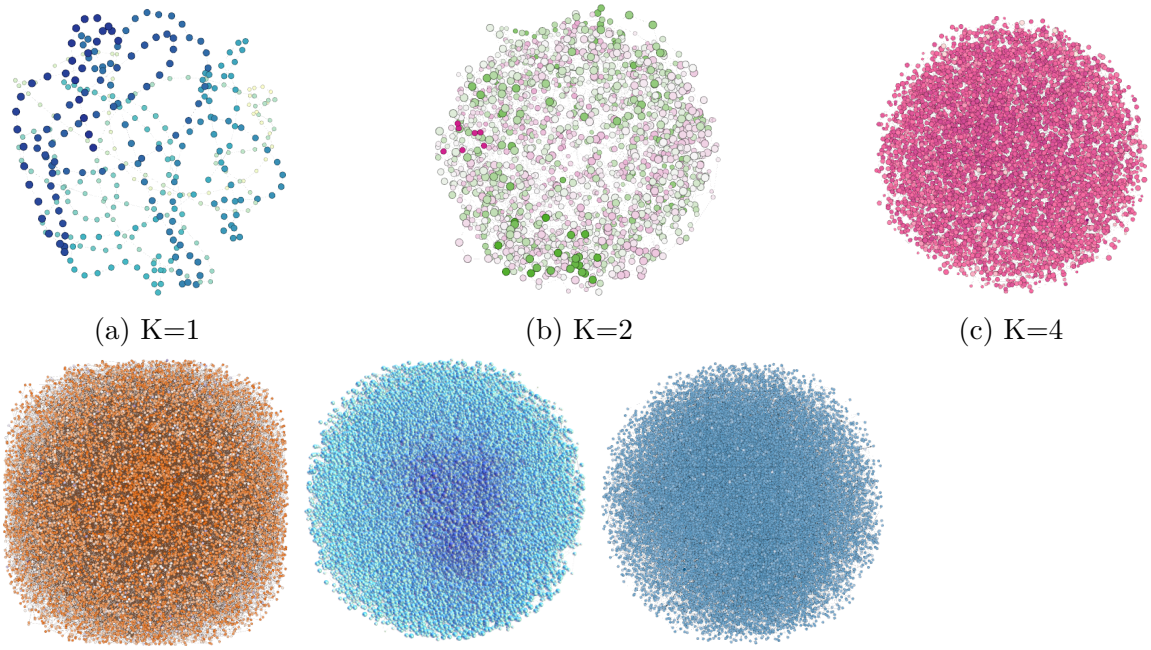
Figure A.25: Betweenness Centrality of sampled graphs for KARWs with 320 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.26: Betweenness Centrality of sampled graphs for NARWs with 320 Khops walk

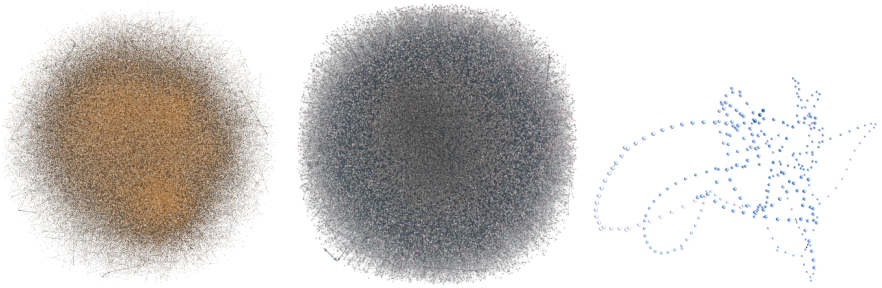
A.18 Closeness centrality of KARWs and NARWs with 320 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

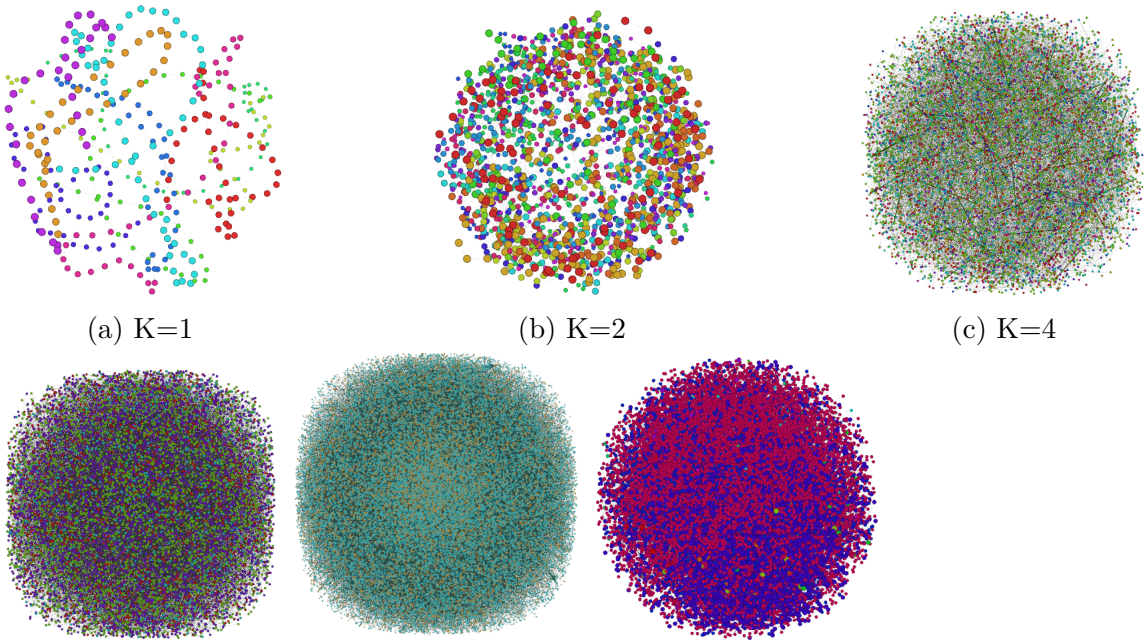
Figure A.27: Closeness Centrality of sampled graphs for KARWs with 320 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.28: Closeness Centrality of sampled graphs for NARWs with 320 Khops walk

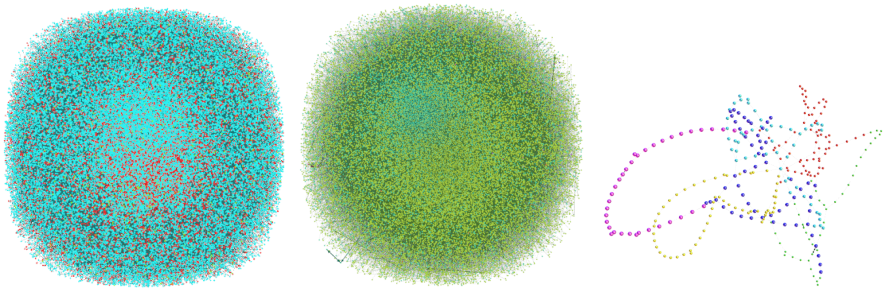
A.19 Modularity of KARWs and NARWs with 320 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

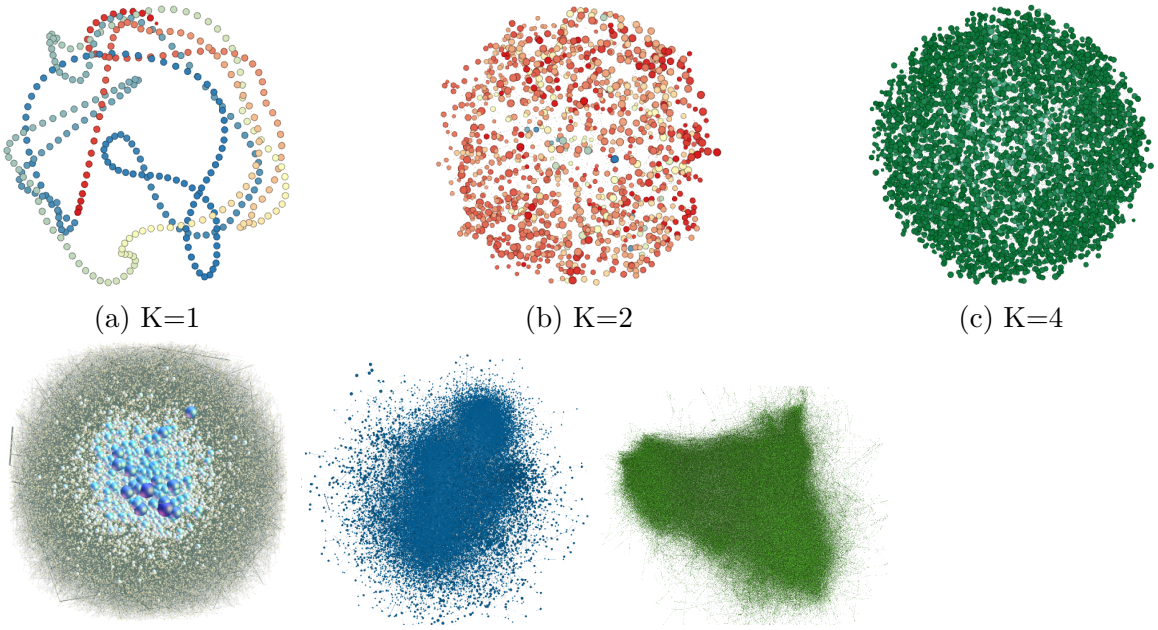
Figure A.29: Modularity of sampled graphs for KARWs with 320 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.30: Modularity of sampled graphs for NARWs with 320 Khops walk

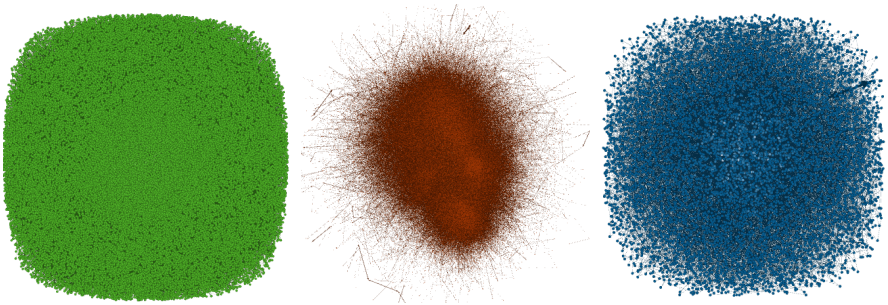
A.20 Betweenness centrality of KARWs and NARWs with 640 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

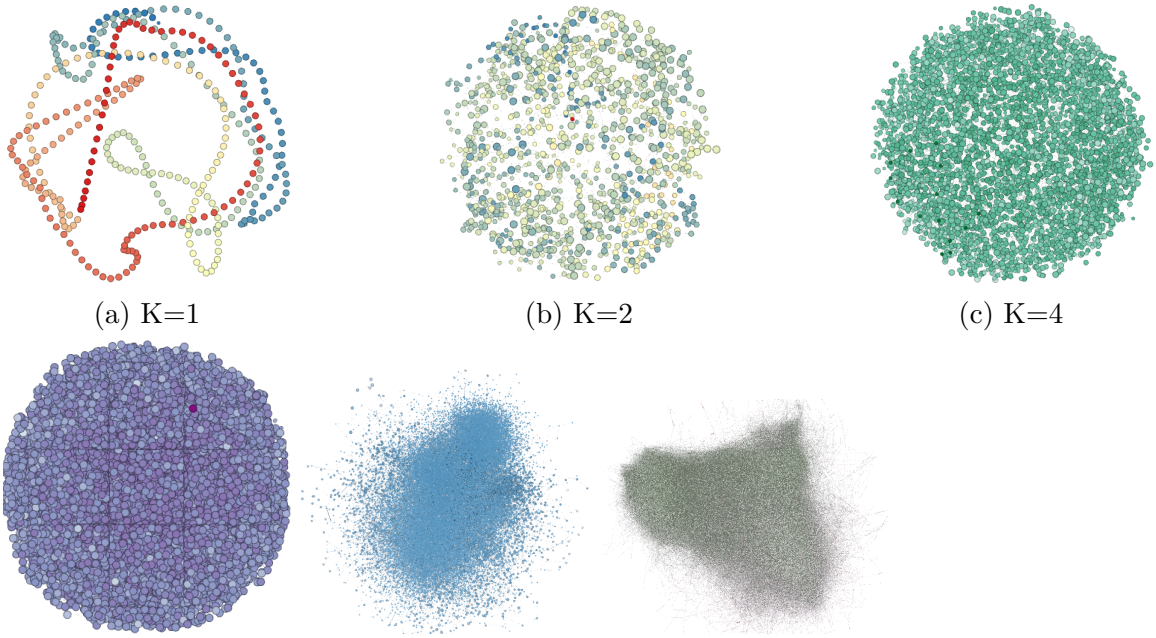
Figure A.31: Betweenness Centrality of sampled graphs for KARWs with 640 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.32: Betweenness Centrality of sampled graphs for NARWs with 640 Khops walk

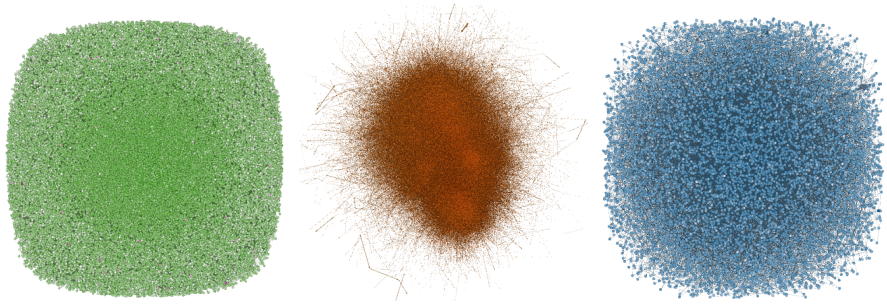
A.21 Closeness centrality of KARWs and NARWs with 640 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

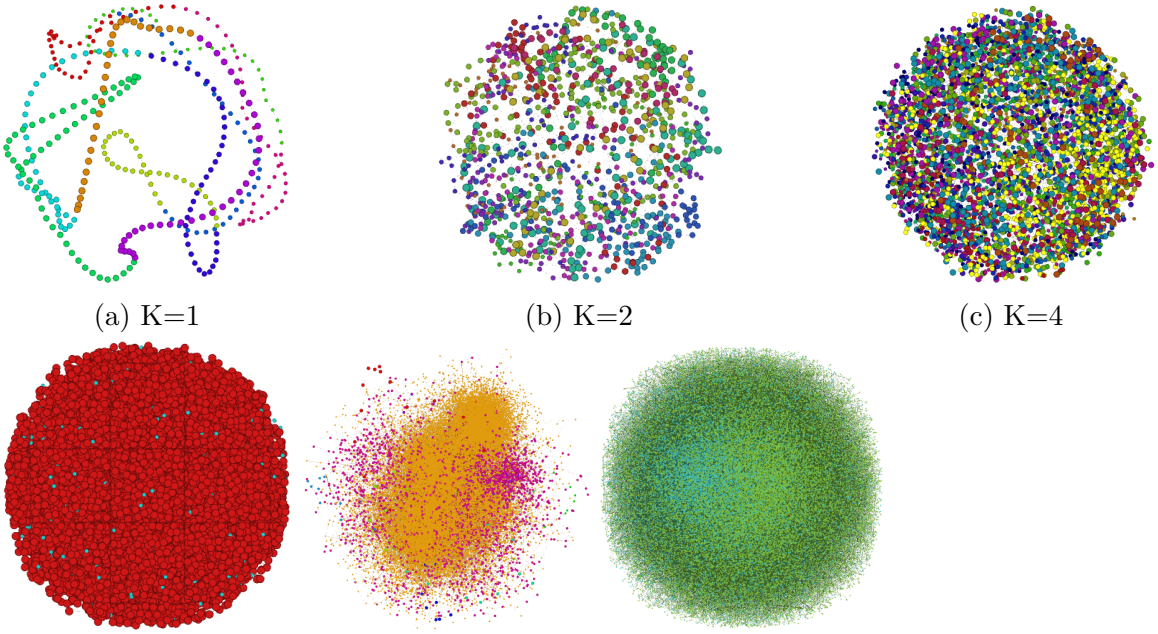
Figure A.33: Closeness Centrality of sampled graphs for KARWs with 640 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.34: Closeness Centrality of sampled graphs for NARWs with 640 Khops walk

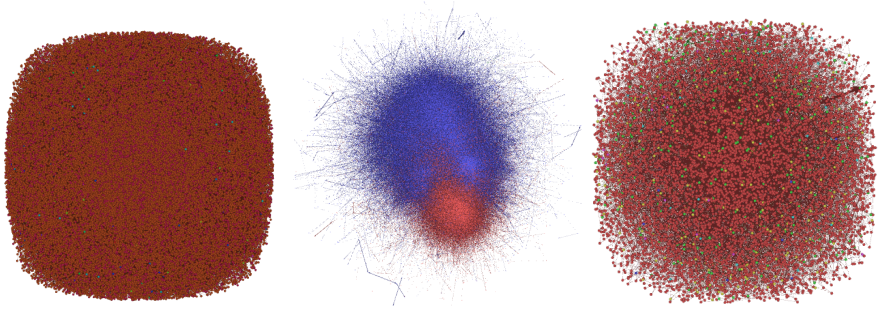
A.22 Modularity of KARWs and NARWs with 640 Khops walk



(a) K=1 (b) K=2 (c) K=4

(d) K=8 (e) K=16 (f) K=32

Figure A.35: Modularity of sampled graphs for KARWs with 640 Khops walk



(a) N=1 (b) N=2 (c) N=4

Figure A.36: Modularity of sampled graphs for NARWs with 640 Khops walk

Bibliography

[1] http://newsroom.fb.com/company-info/.

[2] https://about.twitter.com/company.

[3] http://thenextweb.com/socialmedia/2015/01/21/2015-worldwide-internet-mobile-social-media-trends-get-376-pages-data/.

[4] http://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/.

[5] Nesreen Ahmed, Jennifer Neville, and Ramana Kompella. Network sampling via edge-based node selection with graph induction, 2011.

[6] Nesreen K. Ahmed, Fredrick Berchmans, Jennifer Neville, and Ramana Kompella. Time-based sampling of social network activity graphs. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, MLG '10, pages 1–9, New York, NY, USA, 2010. ACM.

[7] Lars Backstrom and Jon Kleinberg. Network bucket testing. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 615–624, New York, NY, USA, 2011. ACM.

[8] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. In Eytan Adar, Matthew Hurst, Tim Finin, Natalie S. Glance, Nicolas Nicolov, and Belle L. Tseng, editors, *ICWSM*. The AAAI Press, 2009.

[9] L. Becchetti, C. Castillo, D. Donato, A. Fazzone, and I. Rome. A comparison of sampling techniques for web graph characterization. In *Proceedings of the Workshop on Link Analysis (LinkKDD06), Philadelphia, PA*, 2006.

[10] Fabrício Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgílio Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, pages 49–62, New York, NY, USA, 2009. ACM.

[11] Vincent D Blondel, Jean loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks, 2008.

[12] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[13] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, November 1995.

[14] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Rev.*, 51(4):661–703, November 2009.

[15] James Coleman. Relational analysis: The study of social organizations with survey methods. *Human Organization*, 17(4):28–36, 1958.

[16] Minas Gjoka, Carter T. Butts, Maciej Kurant, and Athina Markopoulou. Multigraph sampling of online social networks. *IEEE J. SEL. AREAS COMMUN. ON MEASUREMENT OF INTERNET TOPOLOGIES*, 2011.

[17] Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. A walk in facebook: Uniform sampling of users in online social networks. *CoRR*, abs/0906.0060, 2009.

[18] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, page 130, March 2004.

[19] Leo A Goodman. Snowball sampling. *The annals of mathematical statistics*, pages 148–170, 1961.

[20] Morris H. Hansen and William N. Hurwitz. On the theory of sampling from finite populations. *Ann. Math. Statist.*, 14(4):333–362, 12 1943.

[21] Douglas D Heckathorn. Respondent-driven sampling: a new approach to the study of hidden populations. *Social problems*, 44(2):174–199, 1997.

[22] Douglas D. Heckathorn. Comment: Snowball versus respondent-driven sampling. *Sociological Methodology*, 41(1):355–366, 2011.

[23] Carlos P. Herrero and Martha Saboyá. Self-avoiding walks and connective constants in small-world networks. *Phys. Rev. E*, 68:026106, Aug 2003.

[24] Maciej Kurant, Athina Markopoulou, and Patrick Thiran. On the bias of BFS. *CoRR*, abs/1004.1729, 2010.

[25] Maciej Kurant, Athina Markopoulou, and Patrick Thiran. Towards unbiased BFS sampling. *CoRR*, abs/1102.4599, 2011.

[26] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[27] R. Lambiotte, J. c. Delvenne, and M. Barahona. Laplacian dynamics and multiscale modular structure in networks. *ArXiv*, 2009.

[28] Matthieu Latapy. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theor. Comput. Sci.*, 407(1-3):458–473, November 2008.

[29] S.H. Lee, P.J. Kim, and H. Jeong. Statistical properties of sampled networks. *Physical Review E*, 73(1):016102, 2006.

[30] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 631–636, New York, NY, USA, 2006. ACM.

[31] Marc Najork. Breadth-first search crawling yields high-quality pages. In *In Proc. 10th International World Wide Web Conference*, pages 114–118, 2001.

[32] M. E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 2005.

[33] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2):155–163, 2009.

[34] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *J. of Graph Alg. and App. bf*, 10:284–293, 2004.

[35] Guido Rossum. Python reference manual. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.

[36] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 1–9, New York, NY, USA, 2010. ACM.

[37] Michael P. H. Stumpf, Carsten Wiuf, and Robert M. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences*, 102(12):4221–4224, 2005.

[38] Daniel Stutzbach, Reza Rejaie, Nick Duffield, Subhabrata Sen, and Walter Willinger. On unbiased sampling for unstructured peer-to-peer networks. *IEEE/ACM Trans. Netw.*, 17(2):377–390, April 2009.

[39] Charalampos E. Tsourakakis, U. Kang, Gary L. Miller, and Christos Faloutsos. Doulion: Counting triangles in massive graphs with a coin. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 837–846, New York, NY, USA, 2009. ACM.

[40] V. Umadevi. Article: Automatic co-authorship network extraction and discovery of central authors. *International Journal of Computer Applications*, 74(4):1–6, July 2013. Full text available.

[41] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, August 2009.

[42] Stanley Wasserman, Katherine Faust, and Dawn Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, November 1994.

[43] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, (393):440–442, 1998.

[44] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P.N. Puttaswamy, and Ben Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European Conference on Computer Systems*, EuroSys '09, pages 205–218, New York, NY, USA, 2009. ACM.