Fair Network Division of Nano-satellite Swarms

Evelyne Akopyan, Riadh Dhaou, Emmanuel Lochin, Bernard Pontet, Jacques Sombrin

Fair Network Division of Nano-satellite Swarms

Evelyne Akopyan
TéSA, IRIT, Université de Toulouse, CNRS, Toulouse INP
Toulouse, France
evelyne.akopyan@tesa.prd.fr

Riadh Dhaou
IRIT, Université de Toulouse, CNRS, Toulouse INP
Toulouse, France
riadh.dhaou@toulouse-inp.fr

Emmanuel Lochin
ENAC
Toulouse, France
emmanuel.lochin@enac.fr

Bernard Pontet
CNES
Toulouse, France
bernard.pontet@cnes.fr

Jacques Sombrin
TéSA
Toulouse, France
jacques.sombrin@tesa.prd.fr

***Abstract*—We address the problem of partitioning a network of nano-satellites to distribute fairly the network load under energy consumption constraints. The study takes place in a context where this swarm of nano-satellites orbits the Moon and works as, but not limited to, a distributed radio-telescope for low-frequency radio interferometry. During an interferometry mission, each nano-satellite collects observation data, then shares them with the other swarm members to compute a global image of space. However, the simultaneous transmission of large volumes of data can cause communication issues by overloading the radio channel, leading to potential packet loss. In this context, we investigate three division algorithms based on graph sampling techniques. We prove that random walk-based algorithms overall perform the best in terms of conservation of graph properties and fairness for group sizes down to 10% of the original graph.**

***Index Terms*—Swarm, Network, Fair division, Graph theory, Graph sampling**

I. Introduction

The study of the low-frequency range (below 100 MHz) is essential for many scientific fields, such as astrophysics for sky mapping and monitoring or the observation of the Dark Ages signals, which are signatures of the very early universe [1], [2]. Until now, the majority of low-frequency radio interferometry instruments are ground-based and provide high-quality observations of space. However, the capabilities of these instruments are limited by ionospheric distortions, terrestrial Radio Frequency Interferences (RFI), as well as a complete reflection of radio waves below 10-30 MHz [3]. Therefore, the cosmic radio signals that are weak in terms of power can be easily altered if not completely masked.

One solution to this problem is to create an interferometer directly in space: the Nano-satellites for a Radio Interferometer Observatory in Space (NOIRE) [4] study investigates and proves the benefit of using a swarm of nano-satellites for low-frequency radio observation in space. A swarm of approximately 100 nano-satellites orbiting the Moon is highly protected from the Earth's RFI and ionosphere influence, and thus appears as a promising solution to the interference problem met by ground-based telescopes.

A. Problem definition

The setup and configuration of a swarm of nano-satellites as a space observatory is a challenging problem in terms of communication, as the network relies solely on wireless Inter-Satellite Links (ISL). It is possible to approximate this system as a Wireless Sensor Network (WSN), with the particularity that the average inter-satellite distance goes up to 50 km (interferometry instrument requirement), making it a very low-density network. Besides, the satellites move at an average speed of 1 to 10 km/s, making the swarm a highly mobile system and adding constraints to the communication model. But most importantly, the system needs to operate in a distributed manner. Indeed, unlike traditional observation satellites that usually carry one or many measuring devices, a swarm of nano-satellites consists of a single instrument distributed over many satellites. Thus, each satellite collects observation data from space, then shares them among the swarm to compute one single image of cosmic rays. This image is a matrix of cross-correlations between all data collected by every satellite and shared among the swarm.

The main challenge is thus to disseminate significant amounts of data within the swarm, which raises major communication issues. First, the simultaneous propagation of several gigabytes of data over a radio channel by each satellite can lead to potential link congestion, delay, and packet loss. Then, the energy consumption of the swarm is proportional to the amount of data shared between the satellites. Therefore, a trade-off has to be made to limit the total energy consumption of the swarm to alleviate these issues and prevent too early energy depletion of the satellites. However, the heterogeneous satellite repartition within the topology of the swarm implies that some satellites are more likely to get overloaded and thus go down faster. It becomes primordial to fairly distribute the network load between each satellite of the swarm by taking into account their failure likelihood. One potential solution to distribute the network load is to divide the original network into distinct sub-networks, which are basically groups of satellites from the original network. The objective is thus to improve the communication performance in terms of network load, throughput and failure resilience by performing a fair network division [5].

B. Motivations

Graph division applied to satellite swarms has been studied before in the Orbiting Low-Frequency Array for Radioastronomy (OLFAR) project. In [6], the authors focus on the relevance of node clustering to minimize the energy consumption related to data sharing and provide a slave-master algorithm that proves to be a good solution for their topology. Intuitively, clustering algorithms would perform poorly for fair graph division, because nodes with similar characteristics would be grouped together instead of distributed among the sub-networks. Therefore, clustering algorithms can be used to optimize the energy consumption, but fair division would perform better.

In this paper, we compare the performance of random selection and exploration algorithms to define the best-suited algorithm for network division. We first characterize the swarm network properties by deriving specific metrics, then compare the performance of three graph division algorithms, namely Random Node Division (RND), Multi-Dimensional Random Walk (MDRW), and Forest Fire Division (FFD). RND is the most straightforward division algorithm and a basis for comparison. We chose to study MDRW and FFD because exploration algorithms are expected to perform the best for sub-network sizes of about 15% of the original network, according to previous research on sampling algorithms [7]. We demonstrate that exploration algorithms outperform random selection algorithms regarding network properties conservation. More precisely, MDRW is the best-performing algorithm for sub-networks sizes down to 10% of the original network.

The rest of the paper is organized as follows: in section II, we describe our model. Next, the experimental evaluation is presented in section III, where our results are also confronted with some notable performances of graph sampling algorithms. Finally, section IV summarizes the essential results and perspectives of the work.

II. Model Description

An interferometry mission in space operates in four steps:

1) Observation: each nano-satellite collects raw data from space.
2) Inter-satellite transmission: each nano-satellite broadcasts its data to the others and gathers the data from the other nano-satellites.
3) Computation: each nano-satellite computes a part of the overall space image by combining the data collected and received from the others.
4) Swarm-to-Earth transmission: the swarm downlinks the computed space image to a base station on Earth.

Let us model the swarm network at each timestamp t as a graph $G_t(N_t, E_t)$ with a set of nodes N_t, representing the nano-satellites, and a set of edges E_t, representing the ISLs.

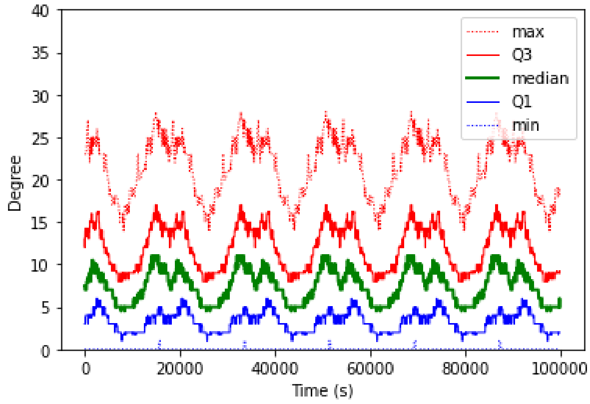


Fig. 1: Evolution of the nodes' degree distribution as a time function. The minimum degree almost always equals 0 despite the variation in the statistical range.

We denote S_G as the set of sub-networks, called subgraphs, of G obtained after division, and T the set of timestamps.

A. Network Properties and Hypotheses

We assume that the nodes of N_0, *i.e.*, in their initial state, are identical regarding power supplies. For all t, the nodes of N_t use omnidirectional antennas to communicate with each other. Then, we assume the model to be faultless, *i.e.*:

$$\forall (i,j) \in T^2,\ |N_i| = |N_j| = |N|$$

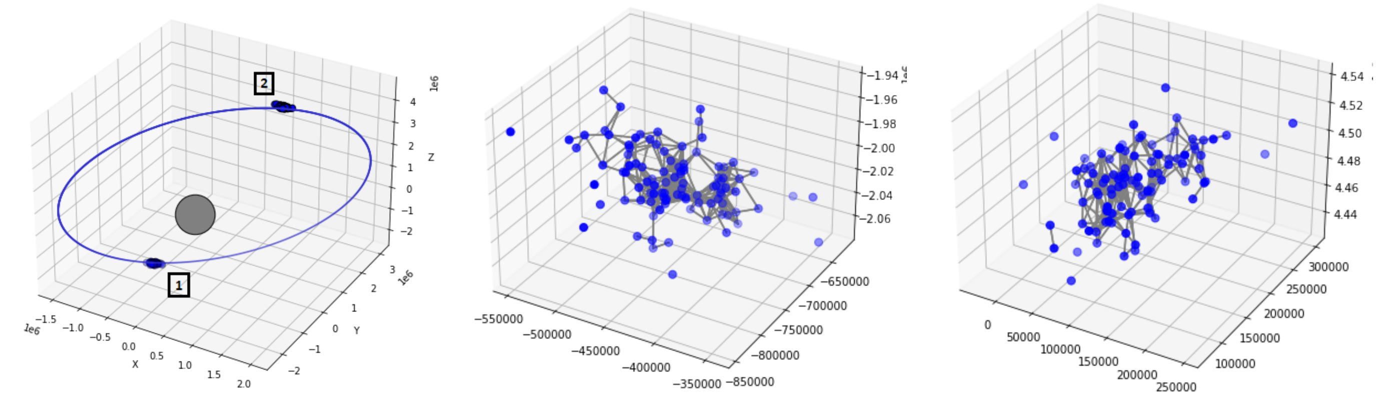
This condition guarantees that the number of nodes in the graph remains constant with time.

Each node must receive at least $\sqrt{|N|}$ data signals to compute $\frac{(|N|-1)}{2}$ cross-correlations [8] to create an image at timestamp t. The global image corresponds to a matrix of $\frac{|N|(|N|-1)}{2}$ cross-correlations between all data signals from each node of N_t. We aim to fairly divide G_t into $|S_G| = \sqrt{|N|}$ subgraphs to restrain network overload during the data-sharing step. Our model has the following properties:

P1: Mobility. The trajectory of the swarm is pseudo-periodic, *i.e.*, a slight random drift is induced in each node's orbit over time. The nodes are also mobile for each other according to a quasi-deterministic mobility model, *i.e.*, $\exists (i,j) \in T^2 \,|\, E_i \neq E_j$. This property highlights the dynamic nature of the swarm network.

P2: Density. The analysis of the distribution of node degrees over time is drawn in Fig. 1, where the minimum, median, and maximum degree values are represented. Q1 and Q3 refer to the 25%-quartile and 75%-quartile, respectively. The figure shows that the network's density is not homogeneous, as high- and low-density zones can appear. Furthermore, the variation in node density is periodic.

P3: Disponibility. The analysis of the inter-contact times provides a measure of the disponibility of the ISLs [9]. Despite an average disponibility below 25%, our model contains a backbone of permanently connected pairs of nodes at all times.



(a) Mobility of the nodes on their lunar orbit. (b) Topology of G at snapshot (1). (c) Topology of G at snapshot (2).

Fig. 2: Dynamic topology of graph G. The orbit of G is defined as the average orbit of its $|N|$ nodes. Each node position is derived with a random offset to the first node position, which is fixed (see Section III-A).

The presence of such backbone implies that these nodes will consume more energy than the rest and thus go down faster.

B. Connection Hypotheses

The connectivity within the swarm is exclusively based on ISLs. Let $e_t(u, v)$ be the edge between two nodes u and v of graph G at time t, and $d(u, v)$ the Euclidean distance between these nodes. The set of edges E_t is defined as follows:

$$E_t = \{e_t(u, v) \,|\, d(u, v) \leq R_G\} \,\forall\, (u, v) \in N_t \qquad (1)$$

where R_G is the connection range of the model: it is independent of time and identical for all nodes in G. We also assume that each ISL is a duplex link, *i.e.*, $e_t(u, v) = e_t(v, u)$.

C. Division Algorithms

First, it is important to highlight the difference between graph division, clustering, partitioning, and sampling. Graph division is a method to split the original graph into smaller subgraphs, with some constraints if specified (*e.g.*, fairness). Graph clustering aggregates nodes in groups (clusters) according to a given similarity. Graph partitioning is a strict form of graph clustering, forcing each node to be part of one and only one group. Graph sampling is a method to create one smaller sample graph that is similar, by some definition, to the original graph. This paper introduces three graph division algorithms based on sampling algorithms. Sampling and division algorithms perform similarly but with different goals: sampling algorithms aim at extracting a single subgraph, while our goal is to divide the graph into multiple subgraphs fairly.

The most straightforward algorithm is the Random Node Division (RND), which is a random selection type of algorithm [7], [10]. Each node of graph G randomly selects a subgraph number in $[0 : |S_G|]$. Nodes with identical subgraph numbers are part of the same subgraph. The algorithm is distributed by nature. The signficant advantage of RND is its complexity of $O(1)$ because the subgraph attribution is instantaneous. We expect this algorithm to work correctly because the number of nodes in our model is high, so the network size should not bias it.

The second algorithm is an exploration algorithm derived from random walks, called Multi-Dimensional Random Walk (MDRW) [10], [11]. We randomly choose $|S_G|$ source nodes, then start random walks in parallel from them (each node selects a random node between its neighbors for propagation). Because our model consists of sparse graphs, a random walk can be stuck if there are no available nodes in its neighborhood. In that case, the node initiates a random jump in the graph to carry on the walk. The algorithm stops when there are no more available nodes in the graph, and the distinct subgraphs consist of the obtained random walks. MDRW is a centralized algorithm, so it has to be distributed in practice. Because there is $|S_G|$ random walks evolving in parallel and the random jump mechanism, the complexity of MDRW is $O(|S_G|)$.

The third algorithm is a forest-fire-based exploration algorithm [7], [10] called Forest Fire Division (FFD). We randomly choose $|S_G|$ source nodes and then start "burning" their neighbors with a probability p. Like MDRW, if a node is stuck when there are still available nodes left in the graph, the node performs a random jump to keep burning elsewhere. The algorithm stops when there are no more available nodes, and the distinct subgraphs consist of the burnt areas. FFD is also a centralized algorithm.

Many other graph sampling algorithms can be adapted for graph division, such as Random Edge sampling, Random Degree Node sampling, Breadth/ Depth/ Random First sampling, or Snow-Ball sampling. However, we specifically selected a subset of them and studied RND, MDRW, and FFD, as these algorithms obtained the best sampling from large graphs [7].

III. Experimental Evaluation

In this section, we first describe the dataset used for the simulation. Then we define the setup of our experiments and the evaluation criteria, then present the results of our experiments.

A. Description of the Dataset

We use synthetic data generated in Matlab, where each node orbits the Moon and follows Kepler's laws, as depicted in Fig. 2a. The trajectory parameters of the first node are set manually on a given orbit. In contrast, the parameters of the others are generated with a random offset to the first one. Fig. 2b illustrates the topology of G_0, *i.e.*, its initial state. The dataset has the following properties:

- the data represent the coordinates of $|N| = 100$ nodes in the Moon-centered coordinate system, distributed in a sphere of 100 km in diameter in its initial state;
- the (x,y,z) coordinates of the nodes are sampled every 10 seconds;
- the simulation duration is $T = 100,000$ seconds.

Geolocalization is impossible to achieve because there is no GPS in outer space. In this case, the Moon-centered coordinate system is convenient, as the nodes process the coordinates to perform a distance-based peer localization.

B. Evaluation Criteria

We evaluate the fairness of the division algorithms by comparing the metrics of the obtained subgraphs with the metrics of the original graph, defined as the reference metrics. Our evaluation is based on five metrics:

- Network size ($|N_S|$): number of nodes in the (sub)graph. $|N_S|$ is constant in time;
- Diameter (Dia): longest shortest path between all pairs of nodes in the (sub)graph, in number of hops;
- Average Degree (AD): average number of neighbors of each node;
- Graph Density (GD): the ratio between the observed number of edges and the maximum possible number of edges in the (sub)graph:

$$GD_t = \frac{2|E_t|}{|N|(|N|-1)}$$

- Average Clustering Coefficient (ACC): for each node, the ratio between the observed number of edges between its neighbors and the maximum possible number of such edges, averaged on the (sub)graph.

Many additional metrics have been proposed in the literature [10], such as the betweenness centrality or assortativity. We choose our set of metrics such that the outputs of the algorithms can be directly compared between them ($|N_S|$ and Dia) and to the original graph (AD, GD, and ACC).

We choose to use Jain's fairness index to compare the metrics obtained with each algorithm, defined for a metric x as follows:

$$J(x) = \frac{1}{1 + c_x^2}$$

where c is the variation coefficient, *i.e.*, the standard deviation over the average ratio of metric x. The best performance is obtained when $J(x)$ tends to 1 for $|N_S|$, AD, GD, and ACC, in other words, when the metrics are properly conserved. The evaluation of Dia is slightly different, as the objective is to minimize its average value and maximize its fairness index simultaneously. Hence, both values are to be taken into account.

C. Simulation Description

We implement our model in Python3 by creating a simple yet adapted simulation module called *swarm_sim*[1]. The setup parameters for our experiment are listed in Tab. I. In our case, time does not need to be taken into account during the graph division because the execution of the algorithm is considerably faster than the time needed for the graph topology and density to evolve.

Variable	Definition	Value
$\|N\|$	Number of nodes	100
$\|S_G\|$	Number of subgraphs	10
R_G	Connection range	30 km
p	Burning probability (FFD)	0.7

TABLE I: Setup parameters of the simulation.

Each algorithm is run 50 times independently. Because random processes are involved in the algorithms, a random seed is generated to impact the output subgraph distributions. For each repetition, the seed takes a different value, and the node distribution of each subgraph is evaluated through the criteria described in III-B. The results of the experiment are summarized in Tab. II, where the best-performing algorithm is bolded for each metric. The reference values are calculated on the original graph G. Then, for each algorithm, we derive the values of each metric averaged on $|S_G|$ subgraphs and on 50 independent repetitions (column $\overline{x}$), and their corresponding Jain's index.

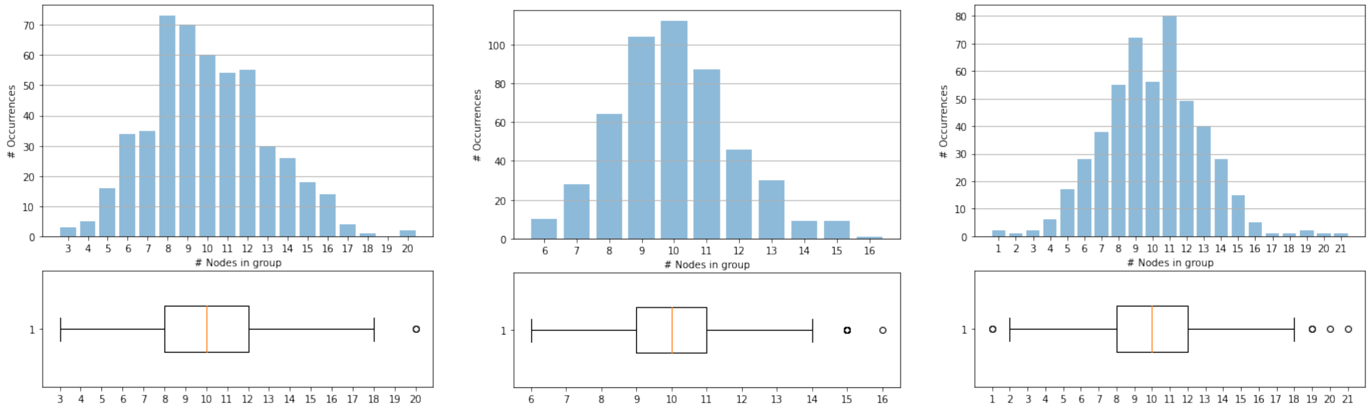
		RND		**MDRW**		FFD	
Metric	Reference	$\overline{x}$	$J(x)$	$\overline{x}$	$J(x)$	$\overline{x}$	$J(x)$
$\|N_S\|$	100	10	0.917	10	**0.967**	10	0.923
Dia	10.2	**6.5**	0.922	6.6	**0.932**	6.6	**0.932**
AD	8.17	8.17	0.954	8.17	0.954	8.17	0.954
GD	0.08	0.08	0.920	0.08	**0.932**	0.08	0.931
ACC	0.5	0.5	0.988	0.5	0.988	0.5	0.988
Average score			0.940		**0.955**		0.945

TABLE II: Fairness performance of RND, MDRW and FFD. MDRW performs the best according to the evaluation criteria.

D. Results Analysis

Surprisingly, we can see from Tab. II, the three algorithms perform very well in conserving metrics. In particular, AD and ACC are equally preserved by RND, MDRW, and FFD, which makes it impossible to make a choice based on these metrics alone. However, the fact that AD and ACC are remarkably well-preserved ($J(x) > 0.95$) implies that one can base its choice of the algorithm on other criteria, such as algorithm complexity. In that case, the best solution would be RND

[1]Link to the source code available upon request.



(a) RND (b) **MDRW** (c) FFD

Fig. 3: Subgraph size distribution for all subgraphs obtained via RND, MDRW and FFD over 50 repetitions of our experiment. MDRW gives the smallest statistical range, resulting in the fairest distribution.

because the the execution of the algorithm is instantaneous, *i.e.* the complexity of RND is $O(1)$. RND also gives the smallest average value of Dia, but its variation is higher.

MDRW and FFD both give acceptable average metrics values. The difference resides in their variation, *i.e.*, their Jain's index: FFD usually has a higher variation than MDRW.

MDRW performs the best overall, outscoring RND and FFD especially in subgraph size, as shown in Fig. 3. Indeed, the probability of getting the fairest node distribution is obtained with MDRW because its statistical range is narrowly centered on $\sqrt{|N|}$. On the other hand, with RND or FFD, getting extremely low or high subgraph sizes are statistically more likely than with MDRW.

Finally, we highlight that only distributed algorithms can be applied to the topology of the swarm (interferometry requirement), so MDRW needs to be adapted accordingly.

E. Related results

Additional results can be found in [7], where the authors compare the performance of graph sampling algorithms (namely Random Node Sampling, Random Walk, and Forest Fire Sampling, among others) in terms of fairness. Although our work focuses on division and not sampling, our results are consistent with theirs. For example, for subgraph sizes of 10% of the original graph, they prove that random walk algorithms perform better than forest fire sampling and random node sampling according to a set of nine network metrics. Furthermore, we obtain the same results when testing with MDRW, FFD and RND and evaluating the performance with only five metrics.

IV. Conclusion

The fair division of a network is primordial to improving the communication performance within a nano-satellite swarm by optimizing energy consumption and enhancing failure resilience. In this paper, we presented the performance of three division algorithms based on graph sampling techniques. We proved that MDRW gives the fairest node distribution although all three algorithms perform fine in terms of properties conservation. MDRW is thus the most adapted to missions where sub-networks need to be very small compared to the original network (10% of the original size). It is essential to highlight that MDRW is the best choice for our setup, but another algorithm might work better if the sub-network sizes need to increase (*e.g.*, in the case of a completely different mission). Nonetheless, the experimental evaluation presented in this paper can be extended to any other swarm network configuration.

Acknowledgments

The authors would like to thank the CNES as well as TéSA laboratory for supporting their work.

References

[1] Brown, L., & Novaco, J. C. (1978). Non Thermal Galactic Emission Below 10MHz. The Astrophysical Journal.

[2] Manning, R., & Dulk, G. A. (2001). The Galactic background radiation from 0.2 to 13.8 MHz. Astronomy & Astrophysics, 372(2), 663-666.

[3] Bentum, M. J., et al (2011). A novel astronomical application for formation flying small satellites. NERG-Nederlands Elektronica en Radiogenootschap, 76(1), 8.

[4] Cecconi, B., et al (2018, March). NOIRE study report: towards a low frequency radio interferometer in space. In 2018 IEEE Aerospace Conference (pp. 1-19). IEEE.

[5] Bentaleb, A., Boubetra, A., & Harous, S. (2013). Survey of clustering schemes in mobile ad hoc networks. Communications and Network, 5(02), 8.

[6] Budianu, A., et al (2011, October). OLFAR: Adaptive topology for satellite swarms. In International Astronautical Congress (pp. 3-7).

[7] Leskovec, J., & Faloutsos, C. (2006, August). Sampling from large graphs. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 631-636).

[8] Corbella, I., et al (2011, July). Interferometric radiometry measurement concept: The visibility equation. In 2011 IEEE International Geoscience and Remote Sensing Symposium. IEEE.

[9] Conan, V., Leguay, J., & Friedman, T. (2007). Characterizing pairwise inter-contact patterns in delay tolerant networks. Autonomics, 7, 19.

[10] Hu, P., & Lau, W. C. (2013). A survey and taxonomy of graph sampling. arXiv preprint arXiv:1308.5865.

[11] Ribeiro, B., & Towsley, D. (2010). Estimating and sampling graphs with multidimensional random walks. In Proceedings of the 10th ACM. SIGCOMM Conference on Internet Measurement (pp. 390–403).