Journal of Computational Science 11 (2015) 34–45

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Node sampling using Random Centrifugal Walks

Andrés Sevilla [a,*], Alberto Mozo [a], Antonio Fernández Anta [b]

[a] Dpto. Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid, Spain
[b] Institute IMDEA Networks, Madrid, Spain

ARTICLE INFO

Article history:
Received 14 January 2015
Received in revised form 1 June 2015
Accepted 2 September 2015
Available online 10 September 2015

Keywords:
Node sampling
Random walks
Randomized algorithms
Distributed algorithms

ABSTRACT

A distributed algorithm is proposed for sampling networks, so that nodes are selected by a special node (source), with a given probability distribution. We define a new class of random walks, that we call Random Centrifugal Walks (RCW). A RCW starts at the source and always moves away from it.

The algorithm assumes that each node has a weight, so the nodes are selected with a probability proportional to its weight. It requires a preprocessing phase before the sampling of nodes. This preprocessing is done only once, regardless of the number of sources and the number of samples taken from the network. The length of RCW walks are bounded by the network diameter.

The RCW algorithms that do not require preprocessing are proposed for grids and networks with regular concentric connectivity, for the case when the probability of selecting a node is a function of its distance to the source.

1. Introduction

Sampling a network with a given distribution has been identified as a useful operation in many contexts. For instance, sampling nodes with uniform probability is the building block of epidemic information spreading [13,14]. Similarly, sampling with a probability that depends on the distance to a given node [3,20] is useful to construct small world network topologies [2,7,16]. Other applications that can benefit from distance-based node sampling are landmark-less network positioning systems like NetICE9 [19], which does sampling of nodes with special properties to assign synthetic coordinates to nodes. In a different context, currently there is an increasing interest in obtaining a representative (unbiased) sample from the users of online social networks [9]. In this paper we propose a distributed algorithm for sampling networks with a desired probability distribution.

1.1. Related work

One technique to implement distributed sampling is to use gossiping between the network nodes. Jelasity et al. [13] implemented a uniform sampling service using gossip-based epidemic algorithms. Kermarrec et al. [15] analyze a generic peer uniform sampling service with small views and independence. Bertier et al. [2] implement uniform sampling and DHT services using gossiping. As a side result, they sample nodes with a distribution that is close to Kleinberg's harmonic distribution (one instance of a distance-dependent distribution). Another gossip-based sampling service that gets close to Kleinberg's harmonic distribution has been proposed by Bonnet et al. [3]. As far as we know, there is no gossip-based sampling algorithm that is able to sample with an arbitrary probability distribution. Moreover, when using gossip-based uniform distributed sampling as a service, it has been shown by Busnel et al. [5] that only partial independence (ϵ-independence) between views (the subsets of nodes held at each node) can be guaranteed without re-executing the gossip algorithm. They show that, in order to achieve ϵ-independence between two consecutive samples at the same node, at least $\Omega(\log(1/\epsilon))$ shuffle rounds must be performed. Each shuffle round involves exchanging $\Theta(n)$ messages (where n is the network size). Gurevich and Keidar [11] give an algorithm that achieves ϵ-independence between uniform samples in $O(ns \log n)$ transformations (i.e., shuffle operations), even in the presence of messages loss, where s is the view size.

Another popular distributed technique to sample a network is the use of random walks [23]. Most random-walk based sampling algorithms do uniform sampling [1,9], usually having to deal with the irregularities of the network. Sampling with arbitrary probability distributions can be achieved with random walks by re-weighting the hop probabilities to correct the sampling bias caused by the non-uniform stationary distribution of the random walks. Lee et al. [17] proposed two new algorithms based on Metropolis–Hastings (MH) random walks for sampling with any probability distribution. These algorithms provide an unbiased

* Corresponding author.
E-mail addresses: asevilla@eui.upm.es (A. Sevilla), amozo@eui.upm.es (A. Mozo), antonio.fernandez@imdea.org (A.F. Anta).

http://dx.doi.org/10.1016/j.jocs.2015.09.001
1877-7503/

graph sampling with a small overhead, and a smaller asymptotic variance of the resulting unbiased estimators than generic MH random walks.

Sevilla et al. [20] have shown how sampling with an arbitrary probability distribution can be done without communication if a uniform sampling service is available. In that work, as in all the previous approaches, the desired probability distribution is reached when the stationary distribution of a Markov process is reached. The number of iterations (or hops of a random walk) required to reach this situation (the warm-up time) depends on the parameters of the network and the desired distribution, but it is not negligible. For instance, Zhong and Sheng [23] found by simulation that, to achieve no more than 1% error, in a torus of 4096 nodes at least 200 hops of a random walk are required for the uniform distribution, and 500 hops are required for a distribution proportional to the inverse of the distance. Similarly, Gjoka et al. [10] show that a MHRW sampler needs about 6K samples (or 1000–3000 iterations) to obtain the convergence to the uniform probability distribution. In the light of these results, Markovian approaches seem to be inefficient to implement a sampling service, specially if multiple samples are desired.

1.2. Contributions

In this paper we present efficient distributed algorithms to implement a sampling service. The basic technique used for sampling is a new class of random walks that we call *Random Centrifugal Walks* (RCW). A RCW starts at a special node, called the *source*, and *always* moves *away* from it.

All the algorithms proposed here are instances of a generic algorithm that uses the RCW as basic element. This generic RCW-based algorithm works essentially as follows. A RCW always starts at the source node. When the RCW reaches a node x (the first node reached by a RCW is always the source s), the RCW stops at that node with a *absorption probability*. If the RCW stops at node x, then x is the node selected by the sampling. If the RCW does not stop at x, it jumps to a neighbor of x. To do so, the RCW chooses only among neighbors that are farther from the source than the node x. (The probability of jumping to each of these neighbors is not necessarily the same.) In the rest of the paper we will call all the instances of this generic algorithm as *RCW algorithms.*

Firstly, we propose a RCW algorithm that samples *any* connected network with *any* probability distribution (given as weights assigned to the nodes). Before starting the sampling, a preprocessing phase is required. This preprocessing involves building a minimum distance spanning tree (MDST) in the network,[1] and using this tree for efficiently aggregating the node's weights. As a result of the weight aggregation, each node has to maintain one numerical value per link, which will be used by the RCW later. Once the preprocessing is completed, any node in the network can be the source of a sampling process, and multiple independent samplings with the exact desired distribution can be efficiently performed. Since the RCW used for sampling follow the MDST, they take at most D hops (where D is the network diameter).

Secondly, when the probability distribution is distance-based and the nodes are at integral distances from the source, RCW algorithms without preprocessing (and only a small amount of state data at the nodes) are proposed. In a *distance-based probability distribution* all the nodes at the same distance from the source node are selected with the same probability. (Observe that the uniform and Kleinberg's harmonic distributions are special cases of distance-based probability distributions.) In these networks, each node at distance $k>0$ from the source has neighbors (at least) at distance $k-1$. We can picture nodes at distance k from the source as positioned on a ring at distance k from the source. The center of all the rings is the source, and the radius of each ring is one unit larger than the previous one. Using this graphical image, we refer the networks of this family as *concentric rings networks*.

Table 1
Success rate of the AAP algorithm as a function of the connectivity angle.

Angle	% success
15°	0%
30°	0%
45°	3%
60°	82%
75°	99%
90°	100%
150°	100%
180°	100%
360°	100%

This concentric rings topology can be naturally found in real networks. For instance, consider a wireless sensor network in which each node has a fixed known position assigned (e.g., via GPS). Then, fixing a source node, the nodes in the kth concentric rings can be the nodes whose (Euclidean) distance to the source is in the interval $(k-1, k]$. If the communication radius is reasonably large, the requirements of the concentric rings topology model will be satisfied.

The first distance-oriented RCW algorithm we propose samples with a distance-based distribution in a network with grid topology. The grid topology has been identified as an efficient deployment pattern in wireless sensor networks [22]. In this network topology, the source node is at position (0, 0) and the lattice (Manhattan) distance is used. This grid contains all the nodes that are at a distance no more than the radius R from the source (the grid has hence a diamond shape[2]). The algorithm we derive assigns an absorption probability to each node, that only depends on its distance from the source. However, the hop probabilities depend on the position (i, j) of the node and the position of the neighbors to which the RCW can jump to. We formally prove that the desired distance-based sampling probability distribution is achieved. Moreover, since every hop of the RCW in the grid moves one unit of distance away from the source, the sampling is completed after at most R hops.

We have proposed a second distance-oriented RCW algorithm that samples with distance-based distributions in concentric rings networks *with uniform connectivity*. These are networks in which all the nodes in each ring k have the same number of neighbors in ring $k-1$ and the same number in ring $k+1$. Like the grid algorithm, this variant is also proved to finish with the desired distribution in at most R hops, where R is the number of rings.

Unfortunately, in general, concentric rings networks have no uniform connectivity. This case is faced by creating, on top of the concentric rings network, an overlay network that has uniform connectivity. In the resulting network, the algorithm for uniform connectivity can be used. We propose a distributed algorithm that, if it completes successfully, builds the desired overlay network. We have found via simulations that this algorithm succeeds in building the overlay network in a large number of cases (see Table 1).

In summary, RCW can be used to implement an efficient sampling service because, unlike previous Markovian (e.g., classical random walks and epidemic) approaches, (1) it always finishes in a number of hops bounded by the network diameter, (2) selects a node with the *exact probability distribution*, and (3) does not need warm-up (stabilization) to converge to the desired distribution.

[1] Using, for instance, the algorithm proposed by Bui et al. [4] whose time complexity is $O(n)$ and $O(n \cdot m)$ message complexity.

[2] A RCW algorithm for a square grid can be derived from the one presented.