Metropolis-Hastings Random Walk with a Reduced Number of Self-Loops

Toshiki Matsumura, and Kazuyuki Shudo
Tokyo Institute of Technology
toshiki7507@gmail.com, and shudo@is.titech.ac.jp

***Abstract*—The random walk (RW) is one of the effective sampling methods for large-scale networks such as the Internet and social networks. While a simple RW method tends to visit nodes with high degrees, the Metropolis-Hastings random walk (MHRW) algorithm has ready-to-use characteristics, wherein the nodes visited by the algorithm distribute uniformly. However, because the MHRW algorithm conducts many self-looping operations and the spread of its crawler is slow, it requires many samples in order to secure sufficient accuracy. This study proposes a RW that requires less self-looping than MHRW did while maintaining the ready-to-use characteristics. The ready-to-use characteristics are achieved by appropriately changing the adaption probability of MHRW and choice probability of neighboring nodes. Our experiments demonstrated that the proposed RW required less self-loop operations than MHRW did. Moreover, the cost of the proposed algorithm is discussed keeping applications to real networks in mind.**

***Index Terms*—graph sampling, Metropolis-Hastings random walk, cost-conscious**

I. Introduction

Graph sampling is a sampling method for data that form a graph structure such as social and traffic networks [1]. Its methodologies can be broadly categorized into two types, namely, random sampling and crawling-based sampling. Random sampling is a method that selects and samples nodes and edges with independent probabilities from the network, whereas crawling-based sampling is a method that selects and samples neighboring nodes and edges. Crawling-based sampling, in particular, is being actively studied because it can be applied to situations that are difficult for random sampling due to restrictions related to security and protection of privacy [2]–[6].

Graph sampling has a diverse range of purposes including the optimization of the calculation and visualization of graphs. For the purpose of accurate estimation of feature values, the random walk (RW) technique is most commonly used. RW is one of the Markov chains, in which the state of the next point is determined solely by the current state. Its probability of visiting each node characteristically converges toward a stationary distribution. As the simplest stationary distribution of RW is proportional to the degree of each node, a sampled node string concentrates on the nodes with high degrees. On the other hand, since its stationary distribution is uniform, the Metropolis-Hastings random walk (MHRW) has ready-to-use characteristics, whereby the expected value of the result of a direct analysis of the sampled node string coincides with the feature value of the population. As is the case with simple RW methods, the MHRW algorithm selects a neighboring node and then decides whether to transit to that node or self-loop to the current node according to the given choice probability. While the MHRW algorithm has the merit of ready-to-use characteristics, it also has demerits such as increase in the number of steps to achieve a certain level of accuracy because it samples the same node repeatedly due to self-looping, or unnecessary costs incurred to learn the degrees of the candidates for transition.

This study proposes a RW method that maintains the ready-to-use characteristics, which is the merit of MHRW, yet has less self-loops than MHRW. The study proves that the proposed method satisfies the ready-to-use characteristics by appropriately changing the adaption probability of MHRW and choice probability of neighboring nodes. In addition, experiments on four networks demonstrated that the number of self-loops in our methods is less than that of the existing MHRW algorithm. The proposed method requires information about two neighboring nodes; therefore, its shortcoming is the difficulty of its application to real networks. In order to overcome it, the degree of the node yet to be visited was approximated using the average degree of the entire network, which allowed us to achieve the same cost as that of the MHRW algorithm. The average degree of the network was estimated from the samples obtained using the MHRW algorithm, which was run for the first few dozen percent of its steps. The sample node string was generated in the remaining steps using this average degree and the proposed method. As the sample node string generated by the MHRW algorithm used to obtain the average degree also follows the uniform distribution, the sample node strings of the MHRW algorithm and proposed method can be combined to estimate the feature value of the population.

The rest of this paper is organized as follows. Chapter 2 will discuss the definitions and description of the terminology used in this study. Chapter 3 will discuss our proposed method. Chapter 4 will discuss the cost required for applying the proposed method to real networks. Chapter 5 will discuss the comparative experiment with MHRW; Chapter 6 will discuss the related analyses. Finally, Chapter 7 will discuss the contributions made by this study and future challenges.

II. Preparation

This chapter defines the symbols used in this paper and premise of graphs. In addition, several algorithms of the RW

Proc. IEEE ISPA/BDCloud/SocialCom/SustainCom 2019,
Decemember 2019

are discussed.

A. Notation

The graph structure is a data structure comprising nodes and edges that connect the nodes. For instance, in a social network service (SNS), its users are the nodes, and the friendships between the users are edges. Thus, it can be analyzed as the graph structure. In this paper, a graph is denoted as $G(V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the node cluster, and $E = \{(v_i, v_j)|v_i, v_j \in V, i \neq j\}$ is the edge cluster. Node number $|V|$ and edge number $|E|$ are set as n and m, respectively. The cluster of nodes neighboring node v_i is defined as $N(v_i) = \{v_j \in V|(v_i, v_j) \in E\}$, while the degree of node v_i is denoted as $d_i = |N(v_i)|$. When the total number of degrees is set to $D = \sum_{i=1}^{n} v_i$, $D = 2E$ is true according to the handshaking lemma. In addition, the average degree is denoted as $\overline{d} = D/n$. Graph G discussed in this paper is an undirected graph, i.e., it has links and no weights, and it is assumed not to have self-loops and multi-edges.

B. Random walk

Graph sampling is a method for elucidating the overall characteristics of an entire graph using as little data as possible [1]. It is effective for large-scale graphs that are expensive to analyze or graphs whose total topology cannot be obtained due to security or privacy issues, which is often the case with social networks. Its methodologies can be broadly categorized into two types, namely, random sampling and crawling-based sampling. In this chapter, one of the crawling-based sampling methods, RW, is discussed. RW is a method where samples are collected using probabilistic transition from the initial node to one of its neighboring nodes. RW is one of the Markov chains. The transition probability P_{ij} from node v_i to node v_j of a simple RW (SRW), which is the simplest RW whose transition probability is uniformly random, can be expressed as follows:

$$P_{ij} = \begin{cases} \dfrac{1}{d_i} & v_j \in N(v_i) \\ 0 & \text{otherwise} \end{cases}$$

The most unique characteristic of RW is that it possesses ergodicity when the following three conditions are satisfied:

1) it is possible to reach from a certain state to any other state;
2) it is not periodic; and
3) the number of states is limited.

When RW possesses ergodicity, stationary distribution π exists simultaneously, and the probability of visiting each node converges toward this stationary distribution. For example, probability distribution $\pi'^{(t)}$ of visiting each node after t steps of SRW can be described as $\pi'^{(t)} = (Pr[x_t = 1], Pr[x_t = 2], ..., Pr[x_t = n])$, where Pr denotes the probability of occurrence of a phenomenon. When i-th element of $\pi'^{(t)}$ is set as $\pi_i'^{(t)}$, it is known that $\pi_i'^{(t)}$ converges toward d_i/D [7]. D is the total number of degrees, and it is a constant. In other words, the probability of visiting each node in SRW is proportional to the nodes' degrees, and the selected samples mainly consist of the nodes with high degrees. In addition, the mixing time is an indicator of the number of steps required until the distribution of samples obtained from RW converges toward a stationary distribution [8]. The mixing time, t_{mix}, can be expressed as follows:

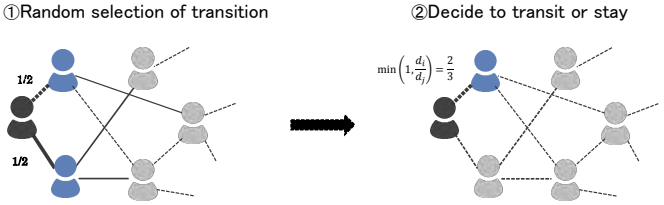

Fig. 1. Example of sampling using the MHRW algorithm. It generates a sample list by repeating ① and ②.

$$\begin{aligned} t_{mix}(\epsilon) &:= \max_{v_i}\{t_{mix}(\epsilon, v_i)\} \\ t_{mix}(\epsilon, v_i) &:= \min_{t}\{|\pi - e_i P^{t'}|_1 \leq \epsilon, \text{for all } t' \geq t\} \end{aligned}$$

where $|\cdot|_1$ is called the total fluctuation distance and expressed according to the following equation:

$$|\pi - e_j P^t|_1 := \frac{1}{2}\sum_{i=1}^{n} |\pi_i - (e_j P^t)_i|$$

The value of parameter ϵ in $t_{mix}(\epsilon)$ is often set to 1/4 or 1/8. The mixing time is closely related not only to the traveling time and total visiting time but also to guarantee of the accuracy of estimation. Hence, it is an important criterion when comparing RW techniques [9]–[12].

1) β-random walk (β-RW): β-RW [13] is a type of RW that was devised to shorten the traveling time and total visiting time of SRW. Its transition probability can be expressed as follows:

Since visiting the nodes with low degrees becomes easier as β becomes larger, it is expected that it would thoroughly visit all nodes of a graph when β is appropriately set unlike the RW technique, which tends to visit nodes with higher degrees. While the expected values of both the traveling time and total visiting time of SRW are $O(n^3)$, it has been proven that the traveling time and the total visiting time of β-RW can achieve $O(n^2)$ and $O(n^2 \log n)$, respectively, when β = 1/2.

2) Metropolis-Hastings random walk (MHRW): MHRW is a RW based on the Metropolis-Hastings (MH) algorithm [14], and it can converge samples toward a chosen distribution. In most cases, it indicates the type where the samples follow the uniform distribution. The transition probability of MHRW can be expressed as follows:

$$P_{ij} = \begin{cases} \dfrac{1}{d_i} \cdot \min(1, \dfrac{d_i}{d_j}) = Q_{ij} \cdot A_{ij} & v_j \in N(v_i) \\ 1 - \displaystyle\sum_{v_k \in N(v_i)} P_{ik} & i = j \\ 0 & \text{otherwise} \end{cases}$$

Algorithm 1 Metropolis-Hastings random walk (MHRW)

Require: n_0: initial node, b: cost
Ensure: $sample\_list$: sample list
  $v_i \leftarrow n_0$
  $sample\_list \leftarrow \{n_0\}$
  $k \leftarrow 1$
  while $k < b$ **do**
    Choose node v_j u.a.r. from $N(v_i)$
    Generate $p \sim U(0,1)$
    if $p \leq \min(1, \frac{d_i}{d_j})$ **then**
      $v_i \leftarrow v_j$
      append v_j to $sample\_list$
    else
      append v_i to $sample\_list$
    end if
    $k \leftarrow k+1$
  end while
  return $sample\_list$

Algorithm 2 Proposed method

Require: n_0: initial node, b: cost
Ensure: $sample\_list$: sample list
  $v_i \leftarrow n_0$
  $sample\_list \leftarrow \{n_0\}$
  $k \leftarrow 1$
  while $k < b$ **do**
    Choose node v_j Q_{ij} from $N(v_i)$
    Generate $p \sim U(0,1)$
    if $p \leq \min(1, \frac{Q_i}{Q_j})$ **then**
      $v_i \leftarrow v_j$
      append v_j to $sample\_list$
    else
      append v_i to $sample\_list$
    end if
    $k \leftarrow k+1$
  end while
  return $sample\_list$

Q and A denote the probability of choosing a neighboring node and adaption probability to decide whether to transit or to stay, respectively. They can be expressed as $Q_{ij} = 1/d_i, A_{ij} = \min(1, d_i/d_j)$.

MHRW algorithm decides whether to transit or to self-loop to the current node using the adaption probability A after selecting a neighboring node using the transition probability of SRW. It adds the neighboring node to the sample list when it transits to that node, and it adds the current node to the sample list again when it self-loops. While the self-looping is one of the characteristics of MHRW algorithm, it also causes degradation of the estimation accuracy due to the increase of the mixing time through multiple sampling of the same node and generation of an extra cost when transition is not made even after obtaining the degree of a transition candidate. MHRW algorithm is shown in Algorithm 1, and its actual sampling flow is illustrated in Figure 1.

III. Proposed method

In the MHRW algorithm, the mixing time is increased due to the self-looping, and it requires many samples to achieve a satisfactory level of accuracy. In this chapter, we propose a RW that maintains the ready-to-use characteristics of the MHRW algorithm while requiring less self-loops compared with the MHRW algorithm.

A. Transition probability

The proposed RW method changes the choice probability of MHRW from that of SRW. Due to this change, the adaption probability is set based on Appendix 1 to satisfy the ready-to-use characteristics. The transition probability of the proposed method can be expressed as follows:

$$P_{ij} = \begin{cases} Q_{ij}A_{ij} & v_j \in N(v_i) \\ 1 - \sum_{v_k \in N(v_i)} P_{ik} & i = j \\ 0 & \text{otherwise} \end{cases}$$

where Q represents the choice probability, and A represents the adaption probability. Unlike the MHRW algorithm that self-loops when it selects a node with a high degree as its transition candidate, the proposed method employs the choice probability that is more likely to select a node when its degree is lower. As a result of reducing the number of the self-loops, the mixing time is reduced and the number of steps required to ensure a satisfactory accuracy is also reduced. The proposed algorithm is shown in Algorithm 2. In this paper, the following two types of choice probabilities are examined.

β-MHRW changes the choice probability of MHRW to that of β-RW. β is set to 1/2, which generates the shortest traveling time and total visiting time, and the choice probability and the adaption probability can be respectively expressed as follows:

$$Q_{ij} = \frac{d_j^{-1/2}}{\sum_{v_k \in N(v_i)} d_k^{-1/2}},$$
$$A_{ij} = \min(1, Q_{ji}/Q_{ij})$$

By employing the choice probability of β-RW, which is proportional to the reciprocal of the degree, the number of self-loops is expected to be reduced.

Reselecting MHRW selects a transition candidate uniformly, and then reselects another transition candidate. Finally, it selects the candidate with a smaller degree. Its choice probability and adaption probability can be respectively expressed as follows:

$$Q_{ij} = \frac{2 \cdot \#\{d_j < d_k\}_{k \in N(i)} + \#\{d_j = d_k\}_{k \in N(i)}}{d_i^2},$$
$$A_{ij} = \min(1, Q_{ji}/Q_{ij})$$

B. Stationary distribution

The transition probability of the proposed method in relation to any of the neighboring nodes is not zero. In addition, as this study postulates that the subject graph is a linked undirected

Algorithm 3 Proposed cost reduction method

Require: n_0: initial node, b: cost, ϕ: Ratio of MHRW
Ensure: $sample\_list$: sample list
  $v_i \leftarrow n_0$
  $sample\_list \leftarrow \{n_0\}$
  $degree\_memory \leftarrow \{n_0\}$: List of nodes whose degrees are obtained
  while length$(degree_memory) < \phi b$ **do**
    Choose node v_j u.a.r. from $N(v_i)$
    if $degree\_memory$ doesn't contain v_j **then**
      append v_j to $degree\_memory$
    end if
    Generate $p \sim U(0,1)$
    if $p \leq \min(1, \frac{d_i}{d_j})$ **then**
      $v_i \leftarrow v_j$
      append v_j to $sample\_list$
    else
      append v_i to $sample\_list$
    end if
  end while
  $average\_degree$ $\leftarrow$ Average degree of the nodes in $sample\_list$
  while length$(degree_memory) < b$ **do**
    Choose node v_j Q_{ij} from $N(v_i)$ (The degree of the node that is not in $degree\_memory$ is calculated as $average\_degree$)
    if $degree\_memory$ doesn't contain v_j **then**
      append v_j to $degree\_memory$
    end if
    Generate $p \sim U(0,1)$
    if $p \leq \min(1, \frac{Q_{ji}}{Q_{ij}})$ (The degree of the node that is not in $degree\_memory$ is calculated as $average\_degree$) **then**
      $v_i \leftarrow v_j$
      append v_j to $sample\_list$
    else
      append v_i to $sample\_list$
    end if
  end while
  return $sample\_list$

graph, it is possible to reach any node from a given node. For the same reason, it is not periodic. Furthermore, based on the premise of the graph, it is clear that the number of nodes is limited. From these three conditions, the proposed RW possesses ergodicity and stationary distribution at the same time. Based on Appendix 1, this stationary distribution is the uniform distribution.

IV. Cost discussion

It is important to consider the cost when conducting sampling. Broadly speaking, there are two ways to consider the cost of MHRW. One is to regard the sample size as the cost, and another is to regard obtaining the degrees as the cost [15]. In this chapter, the behavior of the proposed method under each of these cost considerations is discussed.

A. When the sample size is the cost

It is possible to conduct sampling while considering the sample size as the cost when one would like to set a constant number of the nodes to be obtained, or when there is a limited memory for saving the nodes. As the proposed method has less self-loops than the MHRW does, it is inferred that it can sample a wider variety of nodes when the sample size is the same.

B. When obtaining degrees is the cost

It is possible to conduct sampling while considering the sample size as the cost when one would like to set a constant number of the nodes to be obtained, or when there is a limited memory for saving the nodes. As the proposed method has less self-loops than the MHRW does, it is inferred that it can sample a wider variety of nodes when the sample size is the same.

When obtaining degrees is considered to be the cost, the proposed method requires a far higher cost than the MHRW algorithm even for a per step basis. One possible method to reduce the cost of the proposed method up to that of the MHRW algorithm is to estimate the degrees of the nodes up to the ones that are next to the neighboring nodes. While it is possible to know the degree of a node without incurring any cost if it is already obtained and stored in the memory, it is not possible to know the degree of other nodes. Therefore, the proposed method supplements the degree of the node yet to be obtained with the average degree. By repeating a sufficient number of steps, the number of the nodes whose degrees are obtained can be increased and the samples are expected to converge toward the uniform distribution. However, the average degree cannot be accurately derived without knowing the information of the graph overall, and it is unlikely that the information is given beforehand. Thus, a part of the cost is used to conduct sampling using the MHRW algorithm, and the obtained samples are used to estimate the average degree. The unbiased estimator of function f using RW can be expressed as follows [16]:

$$E_\pi(f) := \sum_{v_i \in V} f(v_i)\pi_i \tag{1}$$

where π is the stationary distribution of RW, and $f : V \rightarrow R$ is a function that uses the node as its argument and returns a real number. As the stationary distribution of MHRW is $\pi = \{1/n, \ldots, 1/n\}$, Equation (1) can be expressed as follows:

$$E_\pi(f) := \frac{1}{n} \sum_{v_i \in V} f(v_i) \tag{2}$$

When $f(v_i) = d_i$, Equation (2) represents the average degree. In this way, the average degree can be derived from MHRW.

Following this, sampling using the proposed method, whose cost is reduced by estimating the average degree, is conducted. As the first sample from MHRW satisfies the ready-to-use characteristics, it can also be used as a sample in its original

TABLE I
DATA SET OVERVIEW.

Network	n	m	D/n
BA model	$10,000$	$29,991$	5.998
Facebook	$4,039$	$88,234$	43.691
DBLP	$317,080$	$1,049,866$	6.622
Amazon	$334,863$	$925,872$	5.530

form, in addition to the estimation of the average degree. The algorithm of the proposed method is shown in Algorithm 3.

When obtaining degrees is the cost, another merit of reducing the self-looping emerges. The MHRW obtains the degree of a neighboring node when deriving the adaption probability. When it self-loops, it does not make a transition to that node even though the degree of the transition candidate has been obtained. In other words, there are many nodes which are not included in the samples although their degrees are obtained. The proposed method can reduce such an unnecessary cost by reducing the number of self-loops.

V. EXPERIMENT

In this chapter, we verify, through experiments using various networks, whether the proposed method has less self-loops compared with the MHRW algorithm. Moreover, we verify whether the ready-to-use characteristics of the proposed method is maintained after changing the choice probability and adaption probability of MHRW.

First, we compare the numbers of self-loops of the MHRW algorithm and the proposed method after crawling for the same number of steps, and verify whether the sample distribution converges toward a uniform distribution (Section V-B). Following this, we conduct the same experiment keeping applications to real networks in mind after equalizing the cost of obtaining degrees of the MHRW algorithm and the proposed method (Section V-C).

A. Data set

Networks generated using the Barabási-Albert (BA) model [17], which is one of the generative models for complex networks, and the data sets of real networks published by the Stanford Network Analysis Project (SNAP)[1] are used for the experiment. Facebook represents the sub-network obtained from the SNS of the actual Facebook. Its nodes represent the users and its edges represent the Follow relationships. DBLP represents the network of the co-authorships of academic papers supplied by the DBLP computer science bibliography. Amazon represents the group purchasing network obtained by crawling the Amazon website. Its nodes represent merchandises and its edges represent the group purchasing relationships indicating whether goods are purchased together frequently or not. Table I shows the characteristics of each network.

B. When obtaining degrees is not the cost

First, we compared the numbers of self-loops of the MHRW algorithm and the proposed method after crawling of the same number of steps. Figure 2 shows the results of conducting a trial where the step that is equivalent to 10% of the total node number was performed 1000 times for each network and the average of the self-loop numbers was obtained. These results shows that the number of self-loops of the proposed method is less than that of the MHRW algorithm for every network. Moreover, L^1 distance is used to verify whether the samples of the proposed method are uniformly distributed. The L^1 distance can be expressed as the equation below. The equation means that the closer the L^1 distance is to zero, the closer the distribution of the observed values is to the expected distribution.

$$L^1 \text{ distance} = \sum_{i=1}^{n} |\pi_i - \pi'_i|$$

where π denotes the distribution to be achieved and π' denotes the actually obtained distribution of the samples.

Figure 3 shows the result of plotting the average of the L^1 distance in relation to the step number obtained for 1000 times for each network. It shows that the proposed method not only satisfies the ready-to-use characteristics but also obtains samples generally closer to the uniform distribution compared with the MHRW algorithm. This is because the L^1 distance of the proposed method is always smaller than that of the MHRW algorithm for every network. It is inferred that this is due to the fact that the crawler avoided sampling only the same nodes; hence, it spread further and collected more diverse nodes in response to the reduction of the self-loop number.

C. When obtaining degrees is the cost

Next, we conducted an experiment with the proposed method after reducing its cost and keeping applications to real networks in mind. Here, it was postulated that obtaining the degree of a node for the first time incurs the cost. The obtained degree of the node is stored in the memory. In the proposed method with a reduced cost, the parameter ϕ is used to determine the ratio of the cost to be allocated to the MHRW that obtains the average degree. In this experiment, the number of self-loops and the L^1 distances for ϕ = 0.1, 0.3, and 0.5 were compared with those of the MHRW algorithm. When ϕ = 1.0, it is MHRW. Figure 4 shows the comparison of the number of self-loops for ϕ = 0.1, 0.3, and 0.5 obtained using the proposed method and the MHRW algorithm. This figure shows that the lower the proportion of the MHRW is, the lower the number of self-loops is. Meanwhile, Figure 5 shows the comparison of the number of specific nodes. It shows that the proposed method is able to sample a greater number of more diverse types of nodes even with the same cost compared with the MHRW algorithm as the proportion of MHRW is lowered.

On the other hand, lowering the proportion of the MHRW leads to worsening of the average degree estimation accuracy, which may negatively affect the ready-to-use characteristics of the proposed method. Figures 6 and 7 show the plot of the L^1 distance for each network, which is the average of 1,000 observations, at 2%, 4%, 6%, 8%, and 10% of the cost of the total number of the nodes. According to the figures, β-MHRW

[1]https://snap.stanford.edu/data/

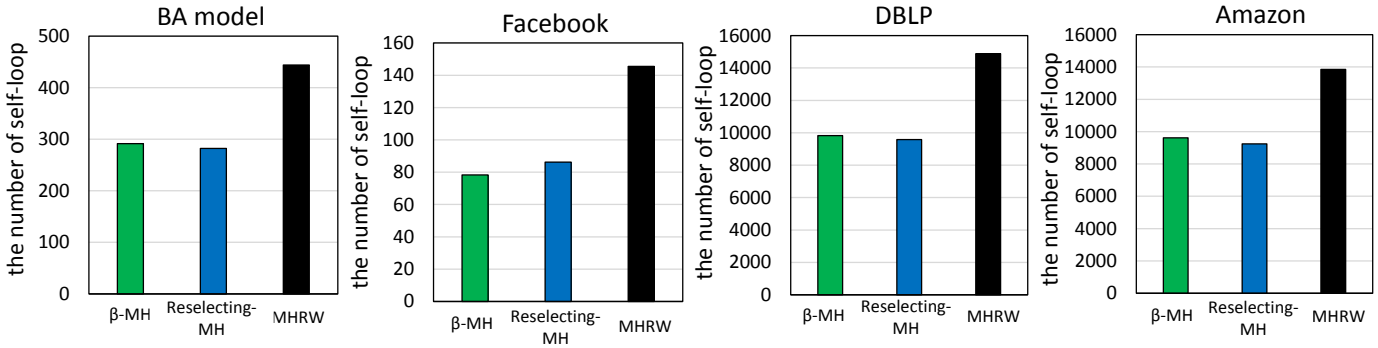

Fig. 2. Number of self-loops of the proposed method and the MHRW algorithm for each network (Step number is 10% of the total number of the nodes).

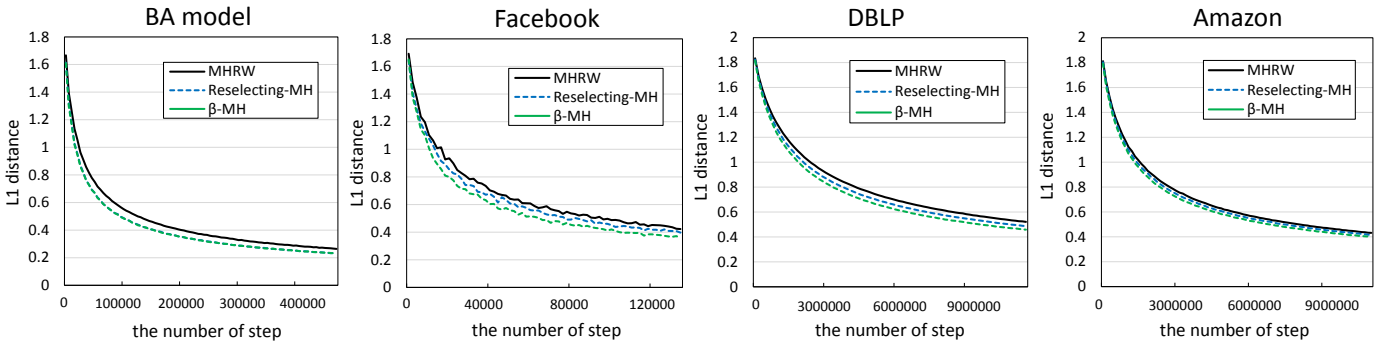

Fig. 3. Transition of the L1 distance for each graph (Horizontal axis is the proportion of the number of steps in relation to the total number of the nodes).

has the shorter L^1 distance than that of the MHRW for every ϕ and every network. In particular, the smaller ϕ is, which is the proportion of the MHRW used to obtain the average degree, the closer the samples of the proposed method are to the uniform distribution. Moreover, the value of L^1 distance of the Reselect MHRW algorithm can be said to be the same or lower than that of the MHRW algorithm. These results suggest that a lower estimation accuracy of the average degree has a small impact on the ready-to-use characteristics of the proposed method.

These experiments indicated that the proposed method is the RW having less number of self-loops than that of the MHRW while maintaining the ready-to-use characteristics. Moreover, it demonstrated that even when the obtaining degrees is the cost, the proposed method can achieve the same cost as the one required by the MHRW algorithm without losing the aforementioned merits when using the proposed cost reduction method.

VI. Related work

Lee et al. proposed an improvement of the MHRW algorithm, called the MH algorithm with delayed acceptance (MHDA), through a different approach from that of this study [18]. While this study reduced the self-loop probability of the MHRW, the MHDA reduces the probability of transition to the previous node. When the MHDA accidentally selected the previous node as the transition candidate, it reselects the transition candidate in a certain probability. As the self-loop probability of the MHDA is equal to that of the MHRW algorithm, it is inferred that combining MHDA with the method proposed in this study would enable the production of more expansive and better RWs.

Chierichetti et al. proposed the maximum-degree sampling (MD) that has the higher number of self-loops compared with that of the MHRW algorithm [19]. When the obtaining degree is the cost, the same node is one expense no matter how many times it is sampled. Thus, if the cost is the same, more samples are collected when the number of the self-loops is higher. The transition probability of the MD is expressed as follows:

$$P_{ij} = \begin{cases} \dfrac{1}{d_{max}} & v_j \in N(v_i) \\ 1 - \dfrac{d_i}{d_{max}} & i = j \\ 0 & \text{otherwise} \end{cases}$$

where $d_{max} = \max\{d_1, d_2, ..., d_n\}$. It has been demonstrated that when obtaining degree is the cost, the L^1 distance of the MD is smaller compared with that of the MHRW algorithm. On the other hand, the cost for obtaining d_{max} under this limited circumstance should be discussed.

VII. Conclusion

This study proposed the RW method having less number of self-loops than that of the existing MHRW while maintaining the ready-to-use characteristics by appropriately changing the choice probability and the adaption probability of the MHRW. The proposed method achieved the reduction of the number of the self-loops by using the choice probability that makes the nodes with smaller degrees more likely to be selected as the transition candidates. Moreover, the proposed method achieved the ready-to-use characteristics by setting the

adaption probability, whose stationary distribution became the uniform distribution, using the MH algorithm.

In order to set the choice and adaption probabilities, the proposed method uses all degrees of the neighboring nodes and the nodes next to them, whereas the MHRW algorithm uses one of the neighboring nodes. Thus, the two methods cannot be said to require the similar cost when applied to the real networks such as SNS and the Internet. Therefore, we devised a method to set the choice and adaption probabilities using the average degree in place of the degrees of the nodes whose degrees are yet to be obtained. Since the situation where the average degree is provided as a prior information would hardly occur, it is estimated first by running the MHRW algorithm using several dozen percent of the cost, and the proposed method is performed using the remaining cost. In this way, the cost of the proposed method was reduced to the similar level of that of the MHRW algorithm.

We compared the number of the self-loops and the ready-to-use characteristics of the proposed method and the MHRW algorithm for four networks in the experiments. The proposed method demonstrated the better results for both the number of the self-loops and the L^1 distance, which is one of the indicators of the ready-to-use characteristics, in the experiment, where the collected sample sizes of both methods were the same. Moreover, we conducted the comparison of the proposed method with a reduced cost and the MHRW algorithms. In the proposed method with the reduced cost, the first 10%, 30%, and 50% of the cost were used to obtain the average degree, and the respective values of the number of the self-loops and the L^1 distance were observed. The results demonstrated that even in the experiments with the same costs, the proposed method has less number of the self-loops, and its sample is closer to the uniform distribution.

Although we used two types of the probability of choosing neighboring nodes in this study, other choice probabilities are possible. We consider that the choice probability that makes the node with the smaller degree more likely to be selected results in the smaller number of the self-loops compared with that of the MHRW algorithms. Therefore, obtaining a better choice probability is a future objective.

ACKNOWLEDGMENT

This work was supported by New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] Pili Hu and Wing Cheong Lau. A survey and taxonomy of graph sampling. *arXiv preprint arXiv:1308.5865*, 2013.

[2] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. ACM KDD 2005*, pages 177–187. ACM, 2005.

[3] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Practical recommendations on crawling online social networks. *IEEE Journal on Selected Areas in Communications*, 29(9):1872–1892, 2011.

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

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

[6] Matthew J Salganik and Douglas D Heckathorn. Sampling and estimation in hidden populations using respondent-driven sampling. *Sociological methodology*, 34(1):193–240, 2004.

[7] David Aldous and Jim Fill. Reversible markov chains and random walks on graphs, 2002.

[8] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[9] Yuval Peres and Perla Sousi. Mixing times are hitting times of large sets. *Journal of Theoretical Probability*, 28(2):488–519, 2015.

[10] Roberto Oliveira et al. Mixing and hitting times for finite markov chains. *Electronic Journal of Probability*, 17, 2012.

[11] Jian Ding, James R Lee, and Yuval Peres. Cover times, blanket times, and majorizing measures. In *Proc. ACM STOC'11*, pages 61–70. ACM, 2011.

[12] Kai-Min Chung, Henry Lam, Zhenming Liu, and Michael Mitzenmacher. Chernoff-hoeffding bounds for markov chains: Generalized and simplified. *arXiv preprint arXiv:1201.0559*, 2012.

[13] Satoshi Ikeda, Izumi Kubo, and Masafumi Yamashita. The hitting and cover times of random walks on finite graphs using local degree information. *Theoretical Computer Science*, 410(1):94–100, 2009.

[14] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[15] Kenta Iwasaki and Kazuyuki Shudo. Comparing graph sampling methods based on the number of queries. In *Proc. IEEE ISPA-IUCC-BDCloud-SocialCom-SustainCom 2018*, pages 1136–1143, 2018.

[16] Rong-Hua Li, Jeffrey Xu Yu, Lu Qin, Rui Mao, and Tan Jin. On random walk based graph sampling. In *Proc. IEEE ICDE 2015*, pages 927–938. IEEE, 2015.

[17] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[18] Chul-Ho Lee, Xin Xu, and Do Young Eun. Beyond random walk and metropolis-hastings samplers: why you should not backtrack for unbiased graph sampling. *ACM SIGMETRICS Performance evaluation review*, 40(1):319–330, 2012.

[19] Flavio Chiericetti, Anirban Dasgupta, Ravi Kumar, Silvio Lattanzi, and Tamás Sarlós. On sampling nodes in a network. In *Proc. WWW 2016*, pages 471–481, 2016.

APPENDIX

Here, we discuss how the adaption probability of the proposed method was set using the MH algorithm. The MH algorithm is one of the representative Markov chain Monte Carlo methods. Its purpose is to compose a sample list under any probability distribution. In order to achieve this objective, the Markov chain, which uniquely possesses a stationary distribution, π, is employed. The stationary distribution and its uniqueness are guaranteed by satisfying the ergodicity and the following detailed balance condition.

$$\pi_i P_{ij} = \pi_j P_{ji} \tag{3}$$

When the stationary distribution is striving to be the uniform distribution, Equation (A-1) becomes $P_{ij} = P_{ji}$. Now, we resolve the transition probability, P, into the choice probability, Q, and the adaption probability, A, namely, $P_{ij} = Q_{ij}A_{ij}$. At this point, the following equation is derived from Equation (A-1).

$$\frac{A_{ij}}{A_{ji}} = \frac{Q_{ji}}{Q_{ij}} \tag{4}$$

When $A_{ij} = \min(1, Q_{ji}/Q_{ij})$, it satisfies Equation (A-2). This is called the Metropolis choice. When $Q_{ij} = 1/d_i$, the transition probability of the MHRW is obtained. The ready-to-use characteristics of the proposed method is satisfied by setting the choice probability of β-RW as Q and the adaption probability of the Metropolis choice as A.

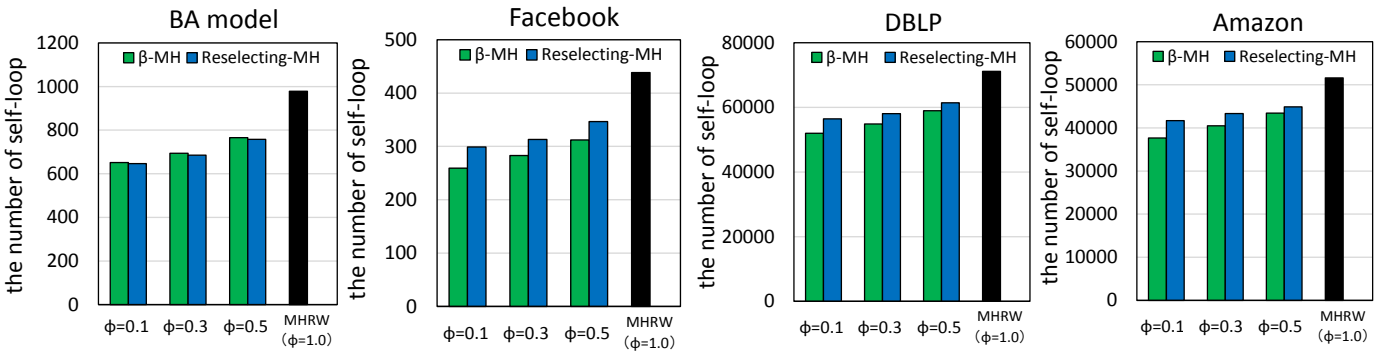

Fig. 4. Number of the self-loops of the proposed method and the MHRW algorithm for each network (Cost is 10% of the total number of the nodes).

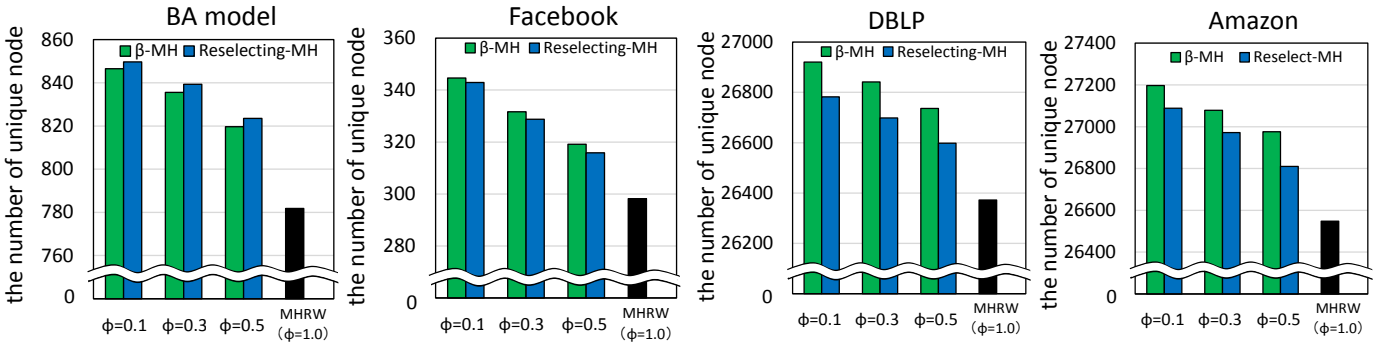

Fig. 5. Number of the specific nodes of the proposed method and the MHRW algorithm for each network (Cost is 10% of the total number of the nodes).

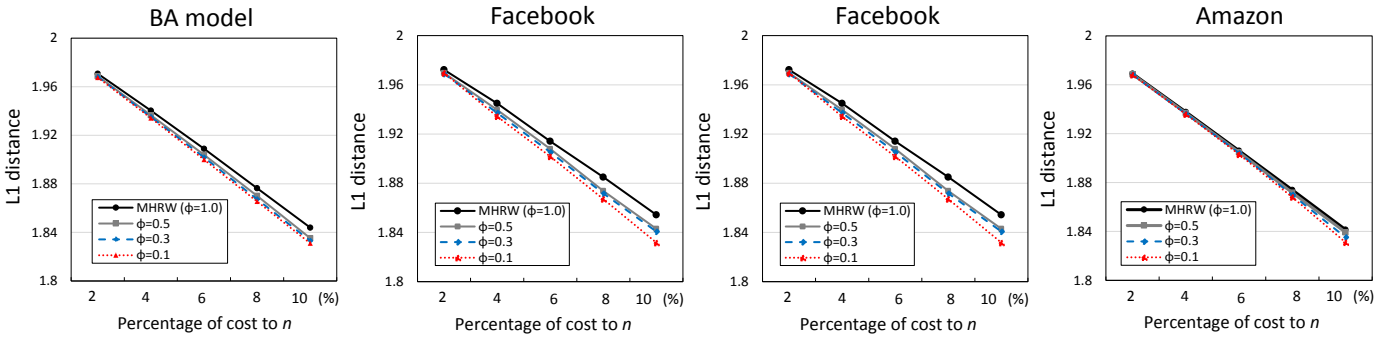

Fig. 6. Transition of the L^1 distance of the MHRW and the β-MHRW algorithms for each graph (Horizontal axis is the proportion of the cost of obtaining degrees in relation to the total number of the nodes).

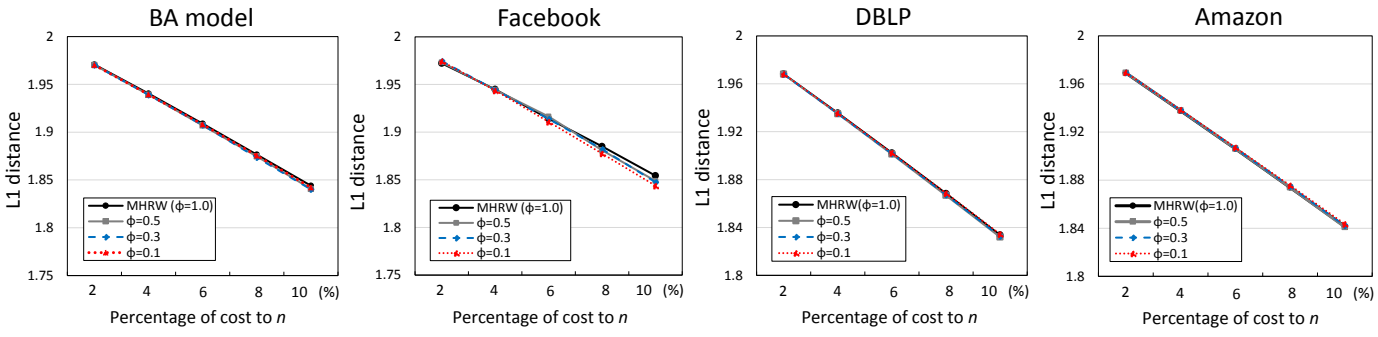

Fig. 7. Transition of the L^1 distance of the MHRW and the Reselecting MHRW algorithms for each graph (Horizontal axis is the proportion of the cost of obtaining degrees in relation to the total number of the nodes).