A Randomized Sampling Algorithm based on Triangle for Community Extraction in Graphs

Yanting Li
School of Information Science and Engineering
Shaoguan University
China
Email:yanting8015@sgu.edu.cn

Ying Huo
School of Information Science and Engineering
Shaoguan University
China
Email:huoying@sgu.edu.cn

***Abstract*—The techniques of sampling play a vital role in extracting communities from graphs. Most of sampling algoritms mainly take the advantage of the degree distribution or the weight of edge. However, it may lead to huge consumption of memory usage and computation time because of the complicated structure of graphs and the noise inherent of algorithm. We propose a novel randomized sampling algorithm, which is a triangle-based sampling algorithm. A random value R_v colors each node uniformly. The edge $e_{ij} = (v_i, v_j)$ is a monochromatic edge if node v_i and v_j receive the same random value R_v. The third edge of triangle Δ_T will be sampled if two edges of the triangle Δ_T are sampled. The triangle Δ_T that formed by three monochromatic edges is considered as the smallest sampling unit in graph G. An extracted community contains at least one triangle. Overviewing, experimental results demonstrate that the proposed algorithm extracts sufficiently dense subgraphs and significantly reduces computation time compared to the reservior sampling algorithm and graph priority sampling algorithm.**

Keywords–Triangle; Monocchromatic edge; Dense subgraph; Pattern extraction; Randomized sampling.

I. INTRODUCTION

The triangle, as the smallest dense subgraph, has gained more and more attention in studying graphs. Triangle is one of the basic shapes of complete subgraphs. Either directed edge or undirected edge, a triangle is the shortest complete cycle and the smallest non-trivial clique. Triangle is applicable to various measurements of network analysis, such as clustering coefficient, transitivity analysis and triangular connectivity. Both transitivity coefficient and clustering coefficient are widely used metrics in the study of social network analysis. Therefore, triangle has emerged as a crucial building block of graphs, thematic structure identification of graphs, node clustering and link classification, etc.

A streaming algorithm has been proposed for counting and sampling triangles from a given graph G [5]. It is one-pass streaming algorithm based on neighborhood sampling. Initially, an edge e_{ij} in the graph stream [6] [31] is randomly extracted. The edge e_{jk} that share the same node n_j with the edge e_{ij} is then extracted. The edge e_{jk} is extracted by employing the neighborhood relationship of edge e_{ij}. By extracting the edges e_{ij} and e_{jk}, the nodes n_i, n_j and n_k can be considered as a potential triangle $T_{(ijk)}$ if the node n_k is the common neighbor node of n_i and n_j. Besides, for studying the characteristic of graphs, a novel notion of dense subgraph named H^*-graph is proposed in [17]. The H^*-graph represents the core of graphs, and extends to encompass the neighborhood among all nodes of the core. Accordingly, an external-memory algorithm for maximal clique enumeration (ExtMCE) is also proposed by employing the H^*-graph for memory usage bounding. Such that, the degree rank of all h-nodes in graph can be computed. The memory usage can be estimated and controlled by using the ExtMCE algorithm. Furthermore, the core of a graph is represented by the subgraph with maximal order of the graph [41], such as k-core of a graph is a core of order k, where $deg_{(v)} \geq k$. The core number of node v is the highest order of the core within maximum value of k [4]. A massive graph is partitioned into a set of subgraphs with smaller size by employing the decomposition algorithm. The characteristic of core graph can be captured after the pruning of sparse components.

To study the characteristic of entire graph, however, becomes expensive due to the increasingly huge size of graph with complicated structure. Numerous sampling techniques have been proposed for extracting essential portions with significant characteristic of graphs. Graph sampling addresses the issue of seeking dense subgraphs, which represent similar properties to the original graph [13] [19] [32] [34]. The reliability of graph sampling techniques is validated by how closely the combinatorial properties of subgraphs simulate the original graph [14]. The interaction network of all proteins that confined to the mitochondria is a real world example. The protein-protein interaction network may not represent the entire network, but can reveal valuable insight into communication or biological process within a defined sphere. Hence, extracting partial of the interaction network of all proteins as the investigation samples is an efficient way of focusing on the sampling property of the network [30].

We propose a novel randomized sampling algorithm, which is based on node coloring for extracting functional unit denoted as Δ_T. A triangle $\Delta_T = \{v_i, v_j, v_k\}$ is considered as the smallest sampling unit, which constitute the community of G. Initially, a random value R_v is uniformly given to each node of a triangle Δ_T in graph G. The edge $e_{ij} = (v_i, v_j)$ is monochromatic if both its two endpoints v_i and v_j receive the same random value R_v. A random value R_v is considered as a color to a node. The third edge is sampled if two edges of a traingle Δ_T are sampled. With the set of monochromatic edges is sampled, all triangles formed by the set of monochromatic

 ISBN: 978-1-61208-806-8

edges can be extracted. Considering triangle as the smallest sampling unit is to prune the search by seeking a few number of densely interconnected communities, which best represent the frequently occurred characteristic of graph G. The random value R_v is a positive value, where $0 \leq R_v \leq |n|$. The $|n|$ is the total number of nodes in G without known beforehand. The random value R_v is unique and unrepeatable for every node. We assume that the range of random values has finite expectations and variances mathematically. We gain the generation of R_v as follow:

$$X_n + 1 = (\frac{{X_n}^2}{10^s})(mod 10^{2s})$$

$$R_v + 1 = \frac{X_n + 1}{10^{2s}}$$

where $(X_n + 1)$ is an iterative operator, and $(R_v + 1)$ is the random value R_v that needs to be generated every time. The s is the shifting of X_n square metre for generating new R_v.

The proposed algorithm aims at extracting communities from graphs. Each edge in graph G is selected based on the uniformly coloring of nodes with probability denoted by P_r, where $0 < P_r < 1$. The color is randomly given to each node, which is a real integer number denoted by R_v. A triangle Δ_T is the smallest sampling unit. A community of graph G contains at least one triangle Δ_T. Therefore, the more triangles anchor in an extracted community, the denser of the ommunity with avaliable insight characteristic of graph G.

This paper is organized as follow. Section I introduces the background of community extraction in graphs, and states the importance of community extraction in network analysis. Section II introduces various approaches of relevant researches. Section III describes the mainframe of randomized sampling algorithm in detail. The experimential results for verifying efficiency of the proposed method are concluded in Section IV. Section V concludes the whole paper.

II. Related Works

A simple and available technique named random sampling has been proposed [8] for scaling the massive graph G into small subgraphs and randomly select samples from the set of subgraphs of G. It is an unbiased sampling technique. Every individual sample is labelled with a random number. The individual sample is randomly extracted from the given matrix according to its labelled random number. The probability of each individual sample being selected at any stage is the same. A systematic sampling method involves the selection of individual samples from an ordered sampling matrix [11]. In this approach, progression through the sample list circles to the top once when the end of the sample list has passed. The sampling procedure begins from selecting an individual sample in the sample list randomly. Every k^{th} individual sample in the sample list is selected where k indicates the sampling interval value. Furthermore, a multistage sampling can be referred in [1] if the matrix data is too expensive to be sampled. This approach is a complicated form of cluster sampling [23]. The clusters are constructed in the given graph G at the first stage. The second stage is to decide the available individual sample in the cluster. All individual samples in the matrix data are appropriately listed. The Random Node Sampling algorithm is a node selection based algorithm [21] [38]. The sampling begins from a given distributed degree of node in a completely self-contained graph G. All nodes of the graph G are given with probability p. The degree distribution of node is denoted by P_k. Extrapolating from the subgraphs to the property of graph G if the randomly extracted subgraph samples have the same characteristic of probability distribution. Alternatively, The TIES algorithm [2] is the total induced edge sampling algorithm. The potential nodes are randomly selected with graph induction based on edge. The degree of node in G is computed. Then, the set of favor nodes with high degrees are selected. The edge e_{ij} is picked up from G at random. The two nodes v_j and v_j are added to the sampled node set in each iteration. The algorithm stops adding nodes to the sampled node set if the fraction $\varnothing$ of nodes are collected. The graph induction process begins once all edges of graph G are traversed. Once all edges that connecting the nodes in the sampled node set, the induced graph is formed. The induced graph holds similar characteristic and structure to the original graph G. A similar sampling approach based on the degree distribution for random graphs is proposed in [38]. The probability of selecting a node is identical for all nodes where $p_i = p$ for all i are considered initial case. The number of connections influences the probability of sampling a node with certain degree is a further sampling scheme for uncorrelated graphs. Therefore, the connectivity of a node denpends on the degrees of its neighbor nodes in the same subgraph. Minne Li, Dongsheng Li and Siqi Shen et al. propose the *DSS algorithm* [27]. All sampling processes is parallelly executed by calculating the exact size of subsample in each partitioning.

The random walk technique is frequently applied onto crawling websites for extracting useful data from the web. The proposal by Bowen and Steve et al. [43] addresses the issue of collecting samples from a graph by adopting random walk. This method achieves the reconstruction of a priori unknown graph. Besides, random walk is a random process technique, in which models the traverse path of a graph by mathematical space. Hence, it is appliable to graph sampling algorithm [7] [35]. An m-dimensional random walk sampling algorithm named Frontier Sampling has been proposed in [7]. The algorithm begins from a set of selected nodes in which preserve the crucial characteristic of regular random walk technique. All nodes of graph G is visited with proportional probability to their degrees. The joint distribution of frontier sampling is similar to the uniform sampling method [40]. Other two graph sampling algorithms, the Rejection-controlled Metropolis-Hashings Algorithm and the Generalized Maximum-degree random walk algorithm, has been proposed in [35]. The Rejection-controlled Metropolis-Hashing Algorithm is a modified Metropolis-Hashing algorithm [40] [44] in which parameterized with an acceptance function α where $0 \leq \alpha \leq 1$. The modified acceptance function improves the acceptance ratio of the original algorithm. Initially, the algorithm begins sampling from a root node v. It stops sampling if the condition of the node v is not satified. Otherwise, node v is selected from its neighbor node w at random. Then, a uniform random value q is generated where $q \in [0, 1]$. The neighbor node w is selected if the uniform random value $q \leq (\frac{d_v}{d_w})^\alpha$. Likewise, the procedure iterates till

the last node with satified parameter in graph G is sampled. Similarly, the generic sampling framework [3] is also based on Metropolis-Hastings algorithm. The notion of the generic sampling is to sample the interesting subgraph patterns without enumerating the entire set of candidate frequent patterns. All candidate patterns form a partial order based on the subgraph relationship. Then, the subgraph samples are returned when the partial order converges to a desired stationary distribution. The function of interestingness of the subgraphs in the sample space determines the stationary distribution selection. The output space sampling is scalable and parallelizable. Besides, the Generalized Maximum-degree random walk algorithm is proposed for unbiased graph sampling. A controlled paramater C is applied onto the original maximum degree algorithm [9] [45] in which C is a nonnegative integer. Similar to the original maximum degree algorithm, the Generalized Maximum-degree random walk algorithm adds $(C - d_v)$ self-loops onto node v of graph G if $d_v < C$ where d_v indicates the degree of node v. Therefore, the walk of GMD algorithm equals to the traditional random walk if $d_v \geq C$. Or the next node that chosen by the GMD walk with probability $\frac{1}{C}$ is the neighbor node of node v. Such that, the round of sampling iteration can be reduced. The graph sampling by random walk begins from a given node and randomly follow the out-connection of the given node [26]. This sampling technique is biased towards the sub-structure of nodes with high connectivity occurrence in a graph. Community extraction is another important method in studying graphs [18] [25]. The partial clustering of nodes in G is computed based on recognizing matrix column similarity. According to the distribution characteristic of graph data, an approach of Horvitz-Thompson estimation to T-stage snowball sampling is proposed by L. C. Zhang and M. Patone [24].

Succinct representation of community in graph is one of the most important techniques in our study. The sampling approach of random multiple snowball with cohen is proposed in [33]. A node is randomly chosen as seed. The neighbor nodes of a root node is selected with the same probability P_c. The process iterates until the set of desirable number of nodes are sampled. Community plays a crucial role in characterizing large-scale complex graphs. A link-tracing sampling algorithm consists of two steps: the set of nodes with shortest path to the set of root nodes is sampled by approximating personalized PageRank vectors, and connect to unvisited neighbor nodes in a new community based on PageRank vectors [28]. The framework of biologicalrelationships are represented by different graph layers It is expected to retain as much information as possible. Didier, Brun and Baudot et al. propose multiplex-modularity approaches to detect communities from multiple graphs [12]. It achieves the recovery of communities more accurately annotated than aggregated counterparts.

III. Randomized Sampling Algorithm

Triangle plays a vital role in both clustering coefficient and transitivity analysis. A triangle consists of three fully connected nodes $\{v_1, v_2, v_3\}$. Either directed edge or undirected edge, a triangle Δ_T can be described as:

$$\Delta_{T_{123}} = \{(v_1, v_2), (v_2, v_3), (v_1, v_3)\}$$

The triangle listing algoirthm is for counting the total number of triangles in a given graph G [37]. The input graph G is iteratively partitioned into a set of subgraphs and stored in the main memory. All triangles in each local subgraph are listed. It is an I/O-efficient in-memory algorithm for extracting subgraphs from G by using a mainframe with limited memory. To list all triangles in a graph could be a huge consumption of memory space. Thus, a decomposition algorithm named truss decomposition [22] is proposed. Similar with k-core decomposition algorithm, truss decomposition algorithm partitions a graph into subgraphs hierarchically. The *k-truss* is defined as the cores of graph G in which every edge of a core must be contained in at least $(k-2)$ triangles. The *k-truss* strengthens each edge in a core by at least $(k-2)$ strong ties. Other notions of subgraphs, such as *k-plex* [36] and *quasi-clique* [20], are proposed for analyzing the social networks. The *k-plex* looses the degree of every node in a clique of c nodes from $(c-1)$ to $(c-k)$. The *quasi-clique* can be considered as a relaxation on the density [16] or the degree [20]. However, the computation of all the above cohesive subgraphs is NP-hard for it could be scattered all over the entire graph, or may overlap largely with eath other. In addition to node-based sampling approaches, the edge-based wedge sampling applies onto estimating the number of triangles in graphs [10].

The approach we proposed considers triangle as the samllest sampling unit due to the full connection among three nodes of a triangle. Contrarily, differ from node-based sampling algorithm [21] and edge based sampling algorithm, the density of extracted subgraph by adopting triangular structure can be moderate. The highly dense components represent the core of the graph G. The key idea of the *Randomized Sampling algorithm* is to color all nodes for extracting a set of monochromatic edges from a graph G. The third edge will be sampled if other two edges of a triangle Δ_T are sampled. A triangle with three monochromatic edges is the smallest sample unit in graph G. The extracted communities of graph G must contain at least one triangle Δ_T. We apply the proposed approach onto social network analysis, as well as the technique of ease that output a sample based on its "closeness" to the original sample [39], or the security system based on biometrics for fingerprint recognition [15].

The randomized sampling algorithm considers the unweighted graph. Given a graph G, the node set and edge set are denoted by V_G and E_G of G respectively. We define $n = |V_G|$ as the number of nodes, and $m = |E_G|$ as the number of edges of G. The size of G is denoted by $|G|$ where $|G| = m + n$. The set of neighbor nodes of node v is denoted by N_v, that is, $N_v = \{u : (u, v) \in E_G\}$.

A triangle Δ_T is a cycle of length 3. Let $\{u, v, w\} \in V_G$ be the three nodes of the cycle. The set of triangles Δ_T is denoted by Δ_G, such that $\Delta_T \in \Delta_G$.

All nodes in graph G will be visited once for sampling monochromatic edges. The searching begins from the root node a in G where $a \in V_G$ as shown in the Fig. 1. The neighbor nodes $\{b,\ c,\ f,\ g\}$ of node a are explored and marked with 1 as visited nodes. The set of nodes $\{a,\ b,\ c,\ f,\ g\}$, however, no triangle anchors inside. Then, the searching continues to seek the neighbor nodes of node b, the set of nodes $\{a,\ h,\ i\}$. The triangle $\Delta_{T_{bhi}}$ is identified for the two nodes $\{h, i\} \in e_{hi}$. The searching strategy is accomplished

by enqueueing each level of the graph G sequentially as the breadth-first search. All neighbor nodes at the present breadth are explored prior to move onto other nodes at next breadth level. The searching stops till the last node in G is visited. Three triangles, $\Delta_{T_{bhi}}$, $\Delta_{T_{cde}}$ and $\Delta_{T_{def}}$ are identified in G. Figure 1 illustrates the path of searching triangles in G.

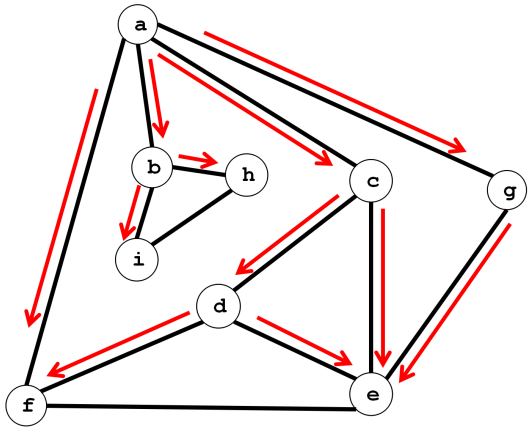

Figure 1: The path of triangle search based on Breadth-First Search

An adjcent list is mapped in Table I.

TABLE I: Adjacent List of nodes in G

node	neighbor nodes
a	b, c, f, g
b	a, h, i
c	a, d, e
d	c, e, f
e	c, d, g
f	a, d, e
g	a, e
h	b, i
i	b, h

Every node is colored once visited. Each node receives a random value denoted by R_v where $0 < R_v < |n|$.

Definition 1 Coloring: The coloring of a node $v \in V_G$, denoted by $cr(v, G)$, is defined as $\{cr(v, G) : R_v$ is uniformly given to each node v, where $0 < R_v < |n|\}$.

Definition 2 Monochromatic edge: $MONO_{e_{uv}}$ denotes the monochromatic edge e_{uv} if $R_u = R_v$. All monochromatic edges are sampled from E_G when all nodes in V_G are colored.

$$MONO_{e_{uv}} = \begin{cases} 1, & \{e_{uv} : R_u = R_v\} \\ 0, & Otherwise \end{cases} \quad (1)$$

Given the graph G=(V_G, E_G), where V_G=$\{a,\ b,\ c,\ d,\ e,\ f,\ g,\ h,\ i\}$. The set of neighbor nodes $\{b, c, f, g\}$ of node a are colored while they are explored. Node $\{a, c\}$ receive red color. Node $\{b, f, g\}$ recieve green color. The $MONO_{e_{ac}}$ can be sampled at the *Stage Two*. At the *Stage Three*, the neighbor nodes of node b, the node $\{h,\ i\}$ receive the green color, the same as node b. Three monochromatic edges, $MONO_{e_{bh}}$, $MONO_{e_{bi}}$ and $MONO_{e_{hi}}$ can be sampled. The iteration process stops till all monochromatic edges are sampled from E_G. Communities containing the $\Delta_{T_{bhi}}$ and $\Delta_{T_{cde}}$ can be extracted from G. Figure 2 illustrates the procedure of node coloring and monochromatic edge sampling.

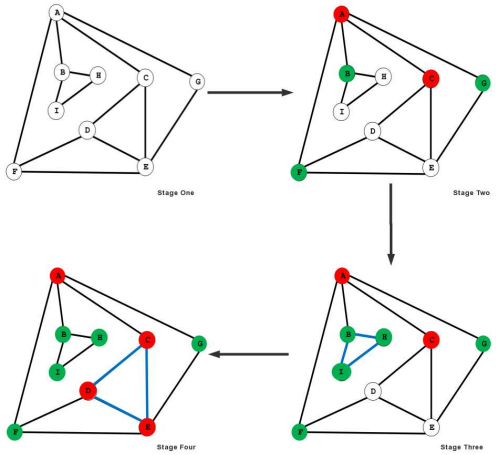

Figure 2: The Sampling of Monochromatic Edges in G

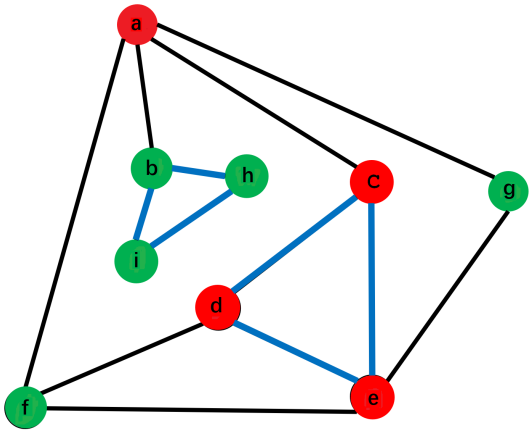

Figure 3: The Extraction of Communities in G

Figure 3 illustrates the extraction of frequent patterns from G. The $\Delta_{T_{def}}$ is not extracted from G as a pattern for e_{df} and e_{ef} are not monochromatic edges.

The randomized sampling algorithm is summarized in algorithm 1. Each node is colored with a random value R_v. Every edge is sampled with probability P_r where $0 \leq P_r \leq 1$.

A. Probability Analysis

1) Global Sampling
The probability of a triangle in G to be extracted as frequent pattern. An edge e_{jk} is monochromatic if its two endpoints j and k receive the same color where $R_j = R_k$. A tiangle that consists of three

Algorithm 1: *Randomized Sampling Algorithm*

- v := node $v \in V$
- e := edge $e \in E$
- R := random value for coloring node
- q := queue for Breadth-First traversal
- $MONO_e$:= the set of monochromatic edges $\in G$
- Δ_G := all triangles of graph G
- Δ_T := triangles contain the set of monochromatic edges

Input: $G = (V_G, E_G)$, R_V = 1,2,3,4....m
Output: a set of triangles Δ_T

```
1  begin
       init R_V, n, q.queue = ∅, Δ_T = ∅
       for i ∈ V_G do
           i.mark = 1
           q.enqueue(v_i)
           R_{V_i} is given to i
           if j is adjacent to i then
               v_i = q.dequeue()
               j.mark = 1
               q.enqueue(v_j)
               R_{V_j} is given to j
               if R_{V_i} = R_{V_j} then
                   e_ij ∈ MONO_e
                   e_ij is sampled
               end
           end
           if k is the common neighbor node of i and j
           then
               v_j = q.dequeue()
               k.mark = 1
               q.enqueue(v_k)
               R_{V_k} is given to k
           end
           Δ_{T_ijk} is identified
           for all triangles Δ_G in G do
               if R_{V_i} = R_{V_j} = R_{V_k} then
                   e_ij, e_jk, e_ik ∈ MONO_e
               end
               Δ_{T_ijk} is extracted
           end
       end
       return a set of triangles Δ_T
   end
```

monochromatic edges $\{e_{ij}, e_{jk}, e_{ik}\} \in MONO_e$ is extracted as a community of G. Then, the triangle $\Delta_{T_{ijk}} \in \Delta_T$. Thus, the probability of a triangle Δ_T to be extracted from G as a pattern is concluded as below.

$$P_r(\Delta_T) = \frac{1}{2} \times \frac{1}{3} = \frac{1}{6}$$

With the increasing numbers of colors which indicated by random values R_v, the probability P_r for every edge to be sampled as a monochromatic edge decreases.

$$P_r = \frac{1}{R_v}$$

2) Local Sampling

The probability of an edge $e \in E_G$ being sampled as a monochromatic edge. For every two nodes j and k that connected by an edge $e_{jk} \in E_G$ receive the same color. Such that, the edge e_{jk} is monochromatic. Then, the $P_r(jk) = P_r(j) \times P_r(k)$. Hence

$$e^2_{(V_G)} = \frac{V_G}{2(k-2)} = \frac{k(k-1)}{2}$$

Then, we obtain the probability P_r for any two nodes $\{j, k\} \in e_{jk}$ where

$$P_r(jk) = e^2_{(V_G)} \times \frac{1}{(V_G)^2} = \frac{(V_G)^2 - V_G}{2} \times \frac{1}{(V_G)^2}$$

B. Complexity Analysis

G is the given graph, let Ψ be the set of candidate frequent patterns contain at least one Δ_T with three monochromatic edges. For $0 \leq R_v \leq |n|$, every node in V is visited once with given a random value R_v. The algorithm initially require $O(|n|scan|G|)$ when giving random value R_v to each node. If the set of $\Delta_T \in \Psi$ of G is extracted, the process requires $O(|m|scan(|\Psi|)) = O(scan(|G|))$ I/O. A monochromatic edge will be removed from Ψ if it does not contained in any triangles. The worst case of the complexity, however, we simply employ an approach with lowest support and extract triangles one by one, which requires the worst case complexity of $O(|n| + |m|scan|G|)$. For the computation of the number of triangles with three monochromatic edges, we at most enumerate all triangles in the corresponding Ψ in which gives $O(\sum_{(\Delta_T \in \Psi)} |m|)$ complexity. Comparatively, the reservior sampling algorithm is a sampling approach that ensure the probability of each element being sampled is the same. The numbers of elements are unknown beforehand. The time complexity of reservior sampling algorithm is $O(n)$ if the total number of being sampled elements is relatively small. However, with increasingly large of the total number of both the elements in the reservior and the set of samples, the time complexity can be up to $O(n*(log(n)-log(n-m)))$. The sampling process uses constant space wiht $O(n*(1+log(m/n)))$ time complexity [42]. The *randomized Sampling algorithm* requires less time complexity than the reservior sampling algorithm when the given graph data is large.

C. Implementation

A $2 * n$ array list is employed for building the storage of all nodes in V_G and their corresponding random values R_v as Figure 4. A continuous storage space must be allocated statically or dynamically. Pointer is set to manage all elements. One is the header pointer in which pointing at the header element. Another is the end-of-list pointer in which pointing at the storage location of the next entry element. The number of elements in the array is constantly changing. Thus, the storage space that occupied by the array moves in the continuous space allocated for the node list. The set of neighbor nodes of node V_a is traversed, but without any anchored triangles. The node V_a and all of its neighbor nodes are removed out of the array for releasing memory space. The set of neighbor nodes of nodes V_b, V_c and V_d contains three triangles so that the set of neighbor nodes is saved to a cluster for storing triangles of G. Other neighbor nodes of nodes V_b, V_c and V_d are visited as

new beginning of traverse process. There is a cut of nodes if it connects to a null in order to prevent from self-loop exploring.

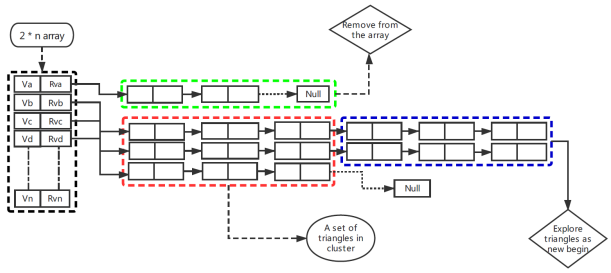

Figure 4: Implememtation of Randomized Sampling Algorithm

IV. Experiment and Evaluation

The performance test of the proposed algorithm is verified via a succession of experiments with the Intel i7 2.3GHz CPU, 16GB RAM, and the version 4.1.2 of C compiler in Mac OS 10.8.3 operating system as the experimential environment.

A. Characteristics of Datasets

Two real world datasets in different sizes are used. The *web-google* dataset is a directed graph, and the *com-LiveJournal* is an undirected graph. The proposed approach fits in processing both directed graph and undirected graph due to the characteristic of triangular structure. Table II shows the features of the two datasets.

TABLE II: FEATURES OF DATASETS

Dataset Statistics	web-Google	com-LiveJournal
$\|V\|$	875,713	3,997,962
$\|E\|$	5,105,039	34,681,189
Clustering Coefficient	0.5143	0.2843
Number of Triangles	13,391,903	177,820,130
Diameter	21	17
Type	directed	undirected

The *web-Google* released by Google is a part of *Google Programming Contest* source in 2002. The *com-LiveJournal* is a free online blogging community. The *com-LiveJournal* dataset provides friendship social network and ground-truth communities. *com-LiveJournal* can create groups when collecting community information according to different features, such as cultural backgroup, lifestyle, technology, entertainment preferences, etc. A community detection technique [18] is inspired by the matrix blocking problem. It is based on the connectivity occurence among all nodes in G. The similarities between a pair of columns in the adjacency matrix is exploited. Two columns in the same block should be more similar than two columns in different block if the patterns are non-zero. A cluster of nodes represents a dense subgraph in G.

B. The Performance of Randomized Sampling Algorithm

The *x-axis* of Figure 5 indicates the selected value of R_v, in which represents the numbers of color. We manully select

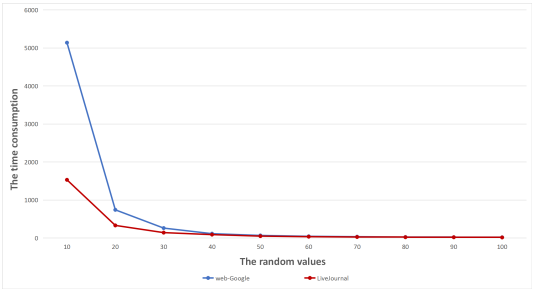

Figure 5: Time Cost of randomized Sampling Algorthm

the set of valuse $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ as random value for each round of communites extraction. The *y-axis* indicates the time consumption counted in second. Throughout the recorded experimental results by using both the datasets of *web-Google* and *com-LiveJournal* in Figure 5, the cost of computation time for communities extraction decreases with the increasing given value of R_v. The time consumption of communities extraction can be analyzed with following two aspects. All nodes are visited once whether each pair of nodes in G receives the same color or not. Therefore, the constant time consumption can be proved by using the Breadth-First Search algorithm for graph traversal. On the other hand, to determine the extracted numbers of communities in G, the process of triangle counting is required. The more counted triangles, the higher time cost, and vice versa.

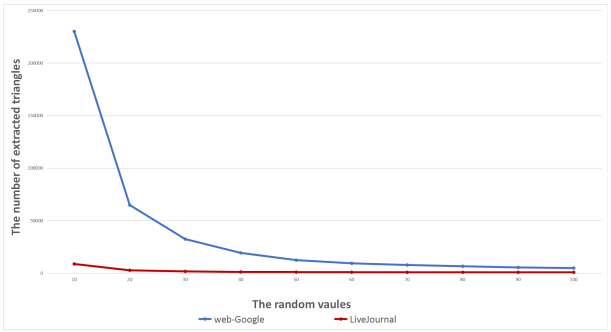

Figure 6: The Number of Extracted Triangles

The extracted triangles that contain the set of monochromatic edges are counted as shown in Figure 6. The *x-axis* indicates the value of R_v. The *y-axis* indicates the number of extracted triangles in G. The chosen values of R_v are the same as shown in Figure 5. Likewise, the numbers of extracted triangles in G decrease with the increasing value of R_v. For the probability of every edge $e \in E_G$ being sampled reduces. Therefore, the number of triangles with three monochromatic edges $e \in MONO_e$ decreases.

For verifying the effciency of the proposed algorithm, both the numbers and the sizes of communities within different values of R_v are examed. Communities in different sizes are extracted. The given graph G can be decomposed hierarchically. The experimental results recorded by Figure 7 proves the distribution results. The *x-axis* indicates the size of communities. The *y-axis* indicates the numbers of communities. The

 ISBN: 978-1-61208-806-8

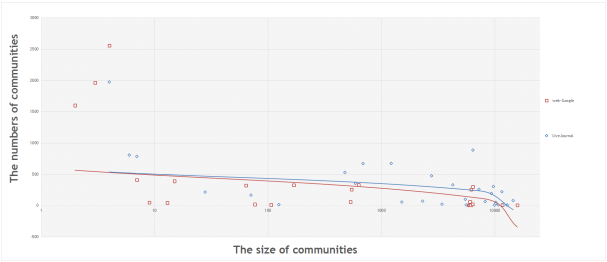

Figure 7: Distribution of Communities

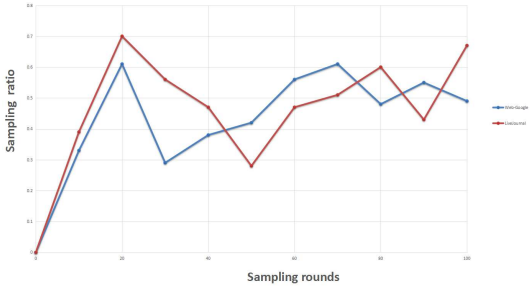

Figure 9: Statistics of Sampling Ratio

red graph shows the distribution of communities of the *web-Google*. The blue graph shows the distribution of communities of the *com-LiveJournal*. With relatively small value of R_v as the sampling threshold, less color for labelling all nodes $n \in N_G$. The probability of edge $e \in E_G$ to be monochromatic becomes higher. A few number of communities in large size are obtained. Contrarily, given relatively large value of R_v as the sampling threshold, more colors for labelling nodes $n \in N_G$. The probability of edge $e \in E_G$ to be monochromatic becomes smaller. Therefore, a large numbers of communities in small size are extracted from G.

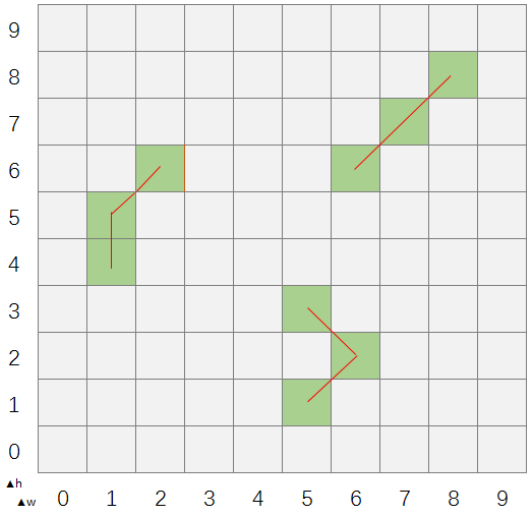

Figure 8: Location of Communities in a graph by using *2-Dimensional Grid*

To locate communities in G, a *2-Dimensional Grid* is proposed as shown in Figure 8. The red graph illustrates the path of community searching. The green blcoks indicate the location of communities in G. It can be computed as the formula below.

$$Grid[h_a][w_a] = \begin{pmatrix} \frac{y_i - y_0}{h_a} + 1 \\ \frac{x_i - x_0}{w_a} + 1 \end{pmatrix}$$

The notion of $\blacktriangle h_a$ and $\blacktriangle w_a$ indicate that community a locates at height of h and width of w respectively.

Figure 9 records the statistic of sampling ratio. The *x-axis* indicates the rounds of sampling process. The *y-axis* indicates the sampling ratio. The results of each sampling can be regarded as a random variable $Var_{(s)}$. Due to the unknown total number of communities in G, and the unpredictable number of extracted samples in the set of communities s, the n rounds sampling results $\{X_1, X_2, X_3, \ldots X_n\} \in s$ can be considered as a set of random variable $Var_{(s)}$. Let $\{X_1, X_2, X_3, \ldots X_n\} \in s$ are the samples selected from the population $G_{(fq)}$, then $\{G_{(fq)} : (X_1, X_2, X_3, \ldots X_n)\}$ is statistical quantity. The sample average can be computed as

$$\bar{Var}_{(s)} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Let the proportion of the amount of selected samples with certain attribute in the population $G_{(fq)}$ defined as the sampling ratio P_s. Then

$$P_s = \frac{s}{G_{(fg)}}$$

We employ the volume of the community for the computation of sampling ratio. Then

$$P_s = \frac{|V_s| + |E_s|}{|V_{G_{(fq)}}| + |E_{G_{(fq)}}|}$$

From the record of Figure 9, we gained remarkable results of sampling ratio both *web-Google* and *com-LiveJournal* at the 20^{th} round.

C. Comparison Experiments

In this experiment, we implement both the reservior sampling algorithm [42] and graph priority sampling algorithm [31] for comparing the performance of communities extraction.

TABLE III: MAXIMUM RUN TIME (IN SECOND)

Dataset	web-Google	com-LiveJournal
Randomized Sampling	5137.18	1529.02
Reservior Sampling	9255.6	6631.07
Graph Priority Sampling	2708.3	3889.032

Table III records the maximum time consumption of sampling. The *Randomized Sampling* is faster than the *Reservior Sampling* in processing both two datasets for the *Randomized Sampling* traverses entire graph G once. Contrarily, the *Reservior Sampling* needs to visit every node in G twice for the

 ISBN: 978-1-61208-806-8

in-degree and out-degree of each node are computed. However, the *Graph Priority Sampling* costs less computation time in extracting communities from web-Google. For *Graph Priority Sampling* (GPS for short) separates the function of edge sampling and sample estimation. The separation of estimation and sampling significantly save resource.

TABLE IV: MAXIMUM NUMBER OF COMMUNITIES

Dataset	Randomized	Reservior	GPS
web-Google	230018	13941	133925
com-LiveJournal	8632	7039	7780

Table IV records the experimental results of maximum numbers of extracted communities by *Randomized Sampling*, *Reservior Sampling* and *Graph Priority Sampling*. The *Randomized Sampling* extractes more communities than the *Reservior Sampling* and the *Graph Priority Sampling* (GPS for short). For a triangle is considered as the samllest community by the *Randomized Sampling*, but a node or an edge cannot be considered as a cohesive subgraph.

TABLE V: DENSITY

Dataset	Randomized	Reservior	GPS
web-Google	0.92	0.85	0.836
com-LiveJournal	0.87	0.69	0.776

Besides, Table V records experimental results of maximum density of communities. We employ the formular below for computating the density of both undirected and directed graphs.

$$dens = \frac{|E_s|}{|V_G| + |E_G|}$$

The results recorded in the table V show that the density of extracted communities by the *Randomized Sampling* are higher than both the *Reservior Sampling* and *Graph Priority Sampling* (GPS for short). The experimential results of the proposed algorithm are competitive and significantly improved.

V. CONCLUSION

We proposed randomized sampling algorithm for extracting communities in graphs. This approach combined the benefits of edge sampling and triangle count to offer high precision of communities extraction. The performance of the randomized sampling algorithm was evaluated based on four measurements, including time consumption, number of triangles, sampling ratio and density of community. Moreover, the superiority of the proposed method was proved by experimental results of comparing with the reservior sampling algorithm and graph priority sampling algorithm. Throughout the experimential results and theoretically analysis, the proposed method was highly confident estimations, and up to ten times sampling size reduction over the state-of-the-art alternatives when the sampling was low.

For the future work, we will prove the analysis of error bound of the randomized sampling algorithm. A modified version of randomized sampling algorithm based on hierachical pruning technique will be proposed.

ACKNOWLEDGMENT

This research is supported by the research fund of Shaoguan University under Grant No.SY2017KJ06 and the Scientific Technology Project of Shaoguan 2019 under Grant No.2019sn063.

REFERENCES

[1] A. Adriana, Gili, J. Elke, Noellemeyer and M. Balzarini, "Hierarchical linear mixed models in multi-stage sampling soil studies," The Environmental and Ecological Statistics, vol.20(2), 2013, pp.237-252.

[2] Ahmed, Nesreen, J. Neville and R. R. Kompella, "Network Sampling via Edge-based Node Selection with Graph Induction," The Computer Science Technical Reports, 2011, paper 1747.

[3] AI Mohammad and M. J. Zaki, "Output Space Sampling for Graph Patterns," Proceedings of VLDB Endowment, vol.2(1), 2009, pp.730-741.

[4] A. Montresor, F. D. Pellegrini and D. Miorandi, "Distributed k-Core Decomposition," IEEE Transactions on Parallel and Distributed Systems, vol.24(2), 2011, pp.288-300.

[5] A. Pavan, K. Tangwongsan, S. Tirthapura and K. L. Wu, "Counting and sampling triangles from a graph stream," The proceedings of the VLDB Endowment, 2013, pp.1870-1881.

[6] A. Zakrzewska and D. A. Bader, "Streaming graph sampling with size restrictions," The IEEE/ACM International Conference, 2017, pp. 282-290.

[7] B. Ribeiro and D. Towsley, "Estimating and sampling graphs with multidimensional random walks," The proceedings of the 10^{th} ACM SIGCOMM conference on Internet measurement ACM, 2010, pp.390-403.

[8] D. David, Chapman and Alexandria, "Selecting Unrestricted and Simple Random with Replacement Samples Using base SAS and PROC SURVEYSELECT," 2012, The SAS Global Forum 2012.

[9] D. Stutzbach, R. Rejaie and N. Duffield, "On unbiased sampling for unstructured peer-to-peer networks," IEEE/ACM Transactions on Networking, vol.17(2), 2009, pp.377-390.

[10] D. Türkoğlu and A. Turk, "Edge-Based Wedge Sampling to Estimate Triangle Counts in Very Large Graphs," 2017, The proceedings of ICDM.

[11] E. Satu and Schaeffer, "Scalable Uniform Graph Sampling by Local Computation," SIAM J. Computer Science, vol.32(5), 2010, pp.2937-2963.

[12] G. Didier, C. Brun, Baudot and Anaïs, "Identifying communities from multiplex biological networks," PeerJ, vol.3(7307), 2015, pp.1525-1545.

[13] G. W. Flake, S. Lawrence and C. L. Giles, "Efficient identification of web communities," The proceediings of the KDD conference, 2000, pp.150-160.

[14] H. Matsuda, T. Ishihara and A. Hashimoto, "Classifying molecular sequences using linkage graph with their pairwise similarities," Theoritical Computer Science, vol.210(2), 1999, pp.305-325.

[15] Ismahane Cheheb, Noor Al-Maadeed, Somaya Al-Madeed, Ahmed Bouridane and Richard Jiang, "Random sampling for patch-based face recognition," The 5_{th} International Workshop on Biometrics and Forensics (IWBF), 2017, pp.1-5.

[16] J. Abello, M. G. C. Resende and S. Sudarsky, "Massive Quasi-Clique Detection," Latin American Symposium on Theoretical Informatics, 2002.

[17] J. Cheng, Y. Ke, A. W. C. Fu, J. X. Yu and L. Zhu, "Finding maximal cliques in massive networks," ACM Transactions on Database Systems, 2011, vol.36(4).

[18] J. Chen and Y. Saad, "Dense subgraph extraction with application to community detection," IEEE Transactions on Knowledge and Data Engineering, vol.24(7), 2012, pp.1216-1230.

[19] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," The proceedings of the SODA conference, 1998, pp.668-677.

[20] J. Pei, D. Jiang and A. Zhang, "On mining cross-graph quasi-cliques," The proceedings of the 11^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2005.

 ISBN: 978-1-61208-806-8

[21] Jure Leskovec and Christos Faloutsos, "Sampling From Large Graphs," The proceedings of KDD, 2006, pp.631-636.

[22] J. Wang and J. Cheng, "Truss Decomposition in Massive Networks," The proceedings of the VLDB Endowment, 2012, vol.5(9), pp.812-823.

[23] J. Zhang, Y. Pei, G. Fletcher and M. Pechenizkiy, "Evaluation of the sample clustering process on graphs," IEEE Transactions on Knowledge and Data Engineering, 2019, pp.(99):1-1.

[24] L. C. Zhang and M. Patone, "Graph sampling," 2017, METRON, Springer, vol.75(3), pp.277-299.

[25] L. Danon, A. Diaz-Guilera, J. Duch and A. Arenas, "Comparing community structure identification," The Journal of Statistical Mechanics: Theory and Experiment, 2005, vol.09.

[26] M. Henzinger, A. Heydon, M. Mitzenmacher and M. Najork, "On near-uniform URL sampling," The proceedings of the 9^{th} International Conference on World Wide Web, 2000, pp.295-308.

[27] Minne Li, Dongsheng Li, Siqi Shen, Zhaoning Zhang and Xicheng Lu, "DSS: A Scalable and Efficient Stratified Sampling Algorithm for Large-Scale Datasets," Network and Parallel Computing. Springer International Publishing, 2019, pp.133-146.

[28] M. Salehi, H. R. Rabiee and A. Rajabi, "Sampling from complex networks with high community structures," The Journal of Chaos, 2012, vol.22(2), pp.2202-2229.

[29] M. P. H. Stumpf, "Sampling properties of random graphs: the degree distribution," The Journal of Physical review. E, Statistical, nonlinear, and soft matter physics, 2005, vol.72(3).

[30] N. Alon and M. Krivelevich, "Testing k-colorability," SIAM J. Discrete Math., vol.15(2), 2002, pp.211-227.

[31] N. K. Ahmed, N. Duffield, T. L. Willke and R. A. Rossi, "On sampling from massive graph streams," Very Large Databases, 2017, vol.10(11), pp.1430-1441.

[32] P.Hu and W.C.Lau, "A survey and taxonomy of graph sampling," arXiv: Social and Information Networks, 2013.

[33] Q. Gao, X. Ding, F. Pan and W. X. Li, "An improved sampling method of complex network," International Journal of Modern Physics C, 2014, vol.25(05).

[34] R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins, "Extracting large-scale knowledge bases from the web," The proceedings of the VLDB conference, 1999, pp.639-650.

[35] Rong-Hua Li, Jeffrey Xu Yu, Lu qin, Rui Mao and Tan Jin, "On random walk based graph sampling," The proceedings of IEEE 31^{th} International conference on Data Engineering, 2015, pp.927-938.

[36] S. B. Seidman and B. L. Foster, "A graph-theoretic generalization of the clique concept," Journal of Mathematical Sociology, vol.6(1), 1978, pp.139-154.

[37] S. Chu and J. Cheng, "Triangle listing in massive networks and its applications," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, vol.6, 2011, pp.672-680.

[38] Stumpf, C. Wiuf and R. M. May, "Subnets of scale-free networks are not scale-free: sampling properties of networks," The proceedings of the National Academy of Sciences of the United States of America, vol.102(12), 2005, pp.4221-4224.

[39] S. Wang, M. Dash and L. T. Chia, "Efficient Sampling: Application to Image Data," Knowledge discovery and data mining, 2005, pp.452-463.

[40] T. Wang, Y. Chen, Z. Zhang and T. Xu, "Understanding Graph Sampling Algorithms for Social Network Analysis," The proceedings of International Conference on Distributed Computing Systems Workshops, 2011, pp.123-128.

[41] V. Batagelj and M. Zaversnik, "An $O(m)$ algorithm for cores decomposition of networks," Advances in data analysis and classification, vol.5(2), 2003, pp.129-145.

[42] Vitter and S. Jeffrey, "Random sampling with a reservoir," ACM Transactions on Mathematical Software, vol.11(1), 1985, pp.37-57.

[43] Y.Bowen, G.Steve, and H.Frank, "Identifying communities and key vertices by reconstructing networks from samples," PLoS ONE, 2013, vol.8(4), e61006-.

[44] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index," The proceedings of International Conference on World Wide Web, 2006, pp.367-376.

[45] Z. Bar-Yossef, A. Berg and S. Chien, "Approximating Aggregate Queries about Web Pages via Random Walks," The proceedings of International Conference on VLDB, 2000, pp.535-544.

 ISBN: 978-1-61208-806-8