# Sampling algorithms for weighted networks

Alireza Rezvanian and Mohammad Reza Meybodi

*Soft computing laboratory, Computer Engineering and Information Technology Department*

*Amirkabir University of Technology (Tehran Polytechnic), Hafez Ave., 424, Tehran, Iran*

## Abstract

Many of the real-world networks, such as complex social networks, are intrinsically weighted networks and therefore traditional network models, such as binary network models will result in losing much of the information contained in the edge weights of the networks and is not very realistic. In this paper, we propose that when the network model is chosen to be a weighted network, then the network measures such as degree centrality, clustering coefficient and eigenvector centrality must be redefined and new network sampling algorithms must be designed to take the weights of the edges of the network into consideration. In this paper, first, some network measures for weighted networks are presented and then, six network sampling algorithms are proposed for sampling weighted networks. The evaluation is done through simulations on real and synthetic weighted networks in terms of relative error, skew divergence, Pearson's correlation coefficient and the Kolmogorov-Smirnov statistic. A number of experiments have been conducted to compare the sampling algorithms for weighted networks proposed in this paper with their counterparts for unweighted networks. The experiments shown that existing sampling algorithms for unweighted networks will not produce good results as used for sampling weighted networks when compared to the algorithms proposed in this paper.

**Keywords:** complex networks, social networks, social network analysis, network sampling, weighted networks.

## 1. Introduction

Many of real systems can be modeled and represented as a network with a set of nodes (*e.g.*, actors of a network) and edges (a kind of connection between actors of networks) (Pálovics and Benczúr 2015; Cordeiro et al. 2016). Although, most of research studies consider networks as binary network, most of the real-world networks, such as online social networks, are intrinsically weighted and the weights in the real-world networks may have arisen in from different aspects. For example in communication networks (Chi et al. 2016), the weight associated with an edge can be the number of calls or the communication time between the cells connected by the links. In citation networks (Zhao et al. 2011), the weight is characterized by the number of citations that the authors give or receive. In online social networks (Jana and Bagchi 2015), the weights of edges can be considered as the number of visits of a user to another user profile, the frequency of interactions among the pairs of users and the rate of friendship activities among users in a community. In transportation networks (Li and Cai 2004), the weight of any given flight is proportional to the degrees of both airports at two ends of that flight. In the biochemical network of metabolic reactions (Saramaki et al. 2005), the nodes are biochemical

elements (enzymes, *etc*) and an edge between two nodes denotes the existence of an individual chemical reaction between them. The weight of an edge can be characterized by the flux of this chemical reaction. Even though it has been widely shown that many real networks are intrinsically weighted, most of the studies on the real networks in the literature have only considered an existence or absence of a connection between nodes which causes the resulting network also called binary networks (or unweighted networks). Modeling the real networks as the binary networks simplifies the metrics, models, algorithms, applications and analyses; however, it will result in losing much of the important information contained in edge weight reflecting the real nature of the network.

In online social networks, the degree of friendship among the users changes directly with the weight of their edges to one another (Rezvanian and Meybodi 2015a) and it is clear that the edges with higher weight are more important, because they are more accessible and willing to communicate or participate in friendship activities, for example, best friends are not treated the same as ordinary friends in online social networks. This issue affects some phenomena, such as information propagation, diffusion of innovations, link prediction and community formations to mention a few. Because the metrics, models and algorithms in binary network are not able to differentiate among users contacted once or multiple times with each other. It suggests that modeling real networks as weighted networks may enhance the network analysis and reinforcement of existing social networks applications. Once the network model is chosen to be a weighted network, metrics, algorithms, applications and analyses must be changed to deal with weighted networks directly.

Network sampling is emerged as a suitable technique to study and analyze real networks. The main purpose of sampling a network is constructing a sampled network with small scale, where the most properties of the original network are preserved. The properties of the networks can be studied and characterized by network sampling with some network measures; these measures of the sampled network are used to study the network instead of the need to access to the whole network data (Murai et al. 2013; Papagelis et al. 2013; Piña-García and Gu 2013; Rezvanian et al. 2014; Jalali et al. 2015; Rezvanian and Meybodi 2015b; Rezvanian and Meybodi 2016b). Using the sampled network obtained from network sampling, one can process information in a small amount of time with lower computational cost (*i.e.*, process, storage, bandwidth, money, *etc.*). Although, several algorithms proposed for network sampling, these sampling algorithms designed solely for binary networks, thus, they are unable to deal with weighted networks directly. Sampling algorithms that use binary network consider just the nodes or edges which result in sampled networks is poor in recovering the important natural characteristics of the original network imbedded in the edge weights.

In this paper, we first present some of the network measures such as degree centrality, clustering coefficient and eigenvector centrality for weighted networks and then, we propose generalization of six network sampling algorithms for sampling weighted networks. In order to study the performance of the sampling algorithm for weighted networks, a number of experiments conducted on real weighted networks in terms of the Kolmogorov-Smirnov (KS) statistic, skew divergence (SD), Pearson's correlation coefficient (PCC), and relative error (RE). The rest of this paper consists of following sections. Section 2 as preliminaries introduces weighted networks, weighted network measures, network sampling and its related work. In section 3, the proposed generalization of six sampling algorithms for weighted networks is described. The performance of the proposed sampling algorithms for weighted networks is studied through the simulation experiments in section 4, and finally section 5 concludes this paper.

## 2. Preliminaries

### 2.1. Weighted networks

A weighted network $G$ is a triple $\langle V, E, W\rangle$ where $V=\{v_1, v_2, \ldots, v_n\}$ is the set of nodes ($n=|V|$), $E=\{e_{ij}\}\subset V\times V$ is the edge-set ($m=|E|$) where $e_{ij}$ indicates a connection between nodes $v_i$ and $v_j$, and $W$ is the set of weights where $w_{ij}$ denotes the weight for edge between nodes $v_i$ and $v_j$. The set of nodes $V$ can represent users in social networks, airline terminals in transportation networks or biochemical elements in biochemical network of metabolic reactions. An edge in $E$ can represent type of a connection, task, activity or flow which can be generated, produced, transferred or terminated through the edges of the network. It needs to be noted that in some applications, the weights are defined according to tasks/activities of network nodes, such as user activities of a user on one's profile in online social networks. Similarly, in some applications, the weights may be defined according to the tasks/activities on an edge between two nodes, such as strength participation between two users in online social networks. In recent years, due to importance of weight in several applications of network analysis, limited studies have been conducted dealing with the weighted networks instead of binary networks. These studies can be categorized into two groups as follows:

I. **Weight as a component for network analysis.** In this approach, weights for the edges in the network are somehow generated by some measures or techniques and then used to improve the performance of the analysis goals. The weight considered for the edge networks are usually generated by structural properties of networks to understand the embedded network structures. An example of this approach is edge prediction in social networks with weighted network. In (Thi et al. 2014), using information flow between two nodes as a similarity, a probability weight is assigned to each edge and the resulting weighed network is used to better task of link prediction. Similar goal with weighting the edge networks for community detection is proposed by *Lu et al.* (Lu et al. 2015). In this study, two new metrics are introduced for edge weighting to characterize the nodes in communities. In (Zheng et al. 2014), in order to study the robustness of network, a strategy for assigning weights based on the concept of the clustering coefficient and betweenness centralities is proposed and the size of the survival components were investigated by simulation. Similar researches with the aim of weighting network by calculating a proximity or similarity between nodes can be found in (Li et al. 2013a; Wang et al. 2013; Zhu et al. 2014; Jarukasemratana and Murata 2015).

II. **Weight as a natural component of network.** A number of studies have attempted to keep weights of connection between edges such as cost, time and volume in their analysis as a vital and natural component of network. Since, several network centralities are incapable of being applied in weighted networks, some network centrality measures have been generalized and developed for weighted networks (Newman 2004; Opsahl et al. 2010). For weighted networks, in (Li et al. 2013b) the edge weights preferential attachment mechanism are used to build a new local-world evolutionary model. Through the simulations, this study shows the structure of realistic networks, such as neural network of the nematode *C. elegans* and online social networks. Since the edge weights can significantly enhance the analysis of network, various applications for weighted networks, such as vulnerability (Dall'Asta et al. 2006), gossip (Tasgin and Bingol 2012), epidemic spreading (Sun et al. 2014) and Identifying influential spreaders (Li et al. 2014) have been investigated by researchers.

Further, several research works tried to study the impact of some beneficial externality of considering or ignoring edge weights and its issues in some applications such as measuring the collaboration

competence in collaboration networks (Yan et al. 2013), link prediction in complex and social networks (Sett et al. 2016), recommendation systems in co-authorship networks (Guns and Rousseau 2014) and analysis of global petroleum exchange in the network of imports and exports of crude petroleum (Yarlagadda et al. 2015) which these studies indicated that the influence of edge weights on the performance of these applications are beneficial. Of course, there are surprisingly some counter examples (Liben-Nowell and Kleinberg 2007; Lü and Zhou 2010) when the edge weights have taken in the algorithm into consideration, as a results, the weighted version of algorithm performed even worse than the unweighted versions. The authors in (Lü and Zhou 2010), discussed about the obtaining weak results when the edge weights have taken into consideration may be related to the theory of well-known weak-ties which claimed that the edges with small weights yet play a more important role in social networks.

## 2.2. Weighted network measures

In this section some famous network measures such as degree centrality, clustering coefficient and eigenvector centrality which are defined for unweighted networks (binary networks) are generalized for weighted networks as given below.

### 2.2.1. Weighted degree (strength)

The degree centrality measure of node $v_i$ in a weighted network also called node strength of node $v_i$ denoted by $wd_i$ is the sum of all the weights along its edges defined as follows (Opsahl et al. 2010)

$$wd_i = \sum_{v_j \in N_i} w_{ij} \tag{1}$$

where $N_j$ is the set of neighboring nodes of node $v_j$ and $w_{ij}$ is the weight of the edges between nodes $v_i$ and $v_j$.

### 2.2.2. Weighted clustering coefficient

The weighted clustering coefficient of node $v_i$ combines the measure of the existence of triples around node $v_i$ with the weight of the edges of $v_i$ which are participating in these triples. The weighted clustering coefficient of node $v_i$ is defined as blow (Saramäki et al. 2007)

$$wc_i = \frac{1}{wd_i(k_i - 1)} \sum_{v_j, v_h \in N_i} \frac{(w_{ij} + w_{ih})}{2} a_{ij}.a_{ih}.a_{jh} \tag{2}$$

where $wd_i$ is the node strength of $v_i$, $k_i$ is the degree of node $v_i$. $a_{ij}$ is 1 if there is an edge between nodes $v_i$ and $v_j$ and is 0 otherwise.

### 2.2.3. Weighted eigenvector

The eigenvector centrality measure node $v_i$ in weighted network denoted by $we_i$ can be defined as follows (Newman 2004)

$$we_i = \frac{1}{\lambda} \sum_{v_j \in N_i} a_{ij}.w_{ij}.e_j \tag{3}$$

where λ is the largest eigenvalue and $e_i = \frac{1}{\lambda}\sum_{v_j \in N_i} a_{ij}.e_j$ is the centrality score for node $v_i$.

## 2.3. Network sampling

Let $G=\langle V, E\rangle$ be the original network, $V$ is the set of nodes and $E$ denotes the edge-set. Network sampling is a sampling method for constructing a sampled network $G'=\langle V', E'\rangle$ from original network $G$ such that $V'\subseteq V$ and $E'\subseteq E$ with sampling rate $0<\varphi<1$, where $|V']=\varphi\times|V|$. It has to be noted that the main goal of network sampling is to achieve a small sampled network while preserving main properties of the original network $G$. Due to the importance of network sampling methods for preprocessing, characterizing, studying and estimating the properties of social networks, several sampling algorithms (Gjoka et al. 2011; Papagelis et al. 2013) were presented for sampling networks that can be categorized into several techniques, such as random selection sampling and traversal based sampling (also called crawling based sampling or topology based sampling). Two simple techniques for random selection sampling are random edge sampling (RES) and random node sampling (RNS) (Leskovec and Faloutsos 2006), which are mainly used for theoretical investigation, regardless of topological structure of the networks. Traversal based sampling algorithms such as Forest Fire Sampling (FFS) (Kurant et al. 2010), Snowball sampling (SBS) (Frank 2011), and Random Walk (RW) (Yoon et al. 2007) try to gather samples from the network with respect to the topological structure of the networks.

### 2.3.1. Related work

In this section, several state of the art algorithms for network sampling in binary networks are reviewed. Some network sampling algorithms have been widely focused on designing accurate and efficient algorithms for characterizing network measures, such as the node degree distribution (Ribeiro and Towsley 2010) and the structure of the groups of the nodes (Kurant et al. 2012; Khomami et al. 2016). A practical framework for uniform sampling from users of *Facebook* has been developed based on crawling method in (Gjoka et al. 2010). In this research, the advantage of unbiased estimation of Metropolis-Hasting random walk (MHRW) sampling and Re-WRW (RWRW) over random sampling and Breadth First Search (BFS) has been addressed with comparing various sampling methods. Cumulative distribution of degrees is estimated via sampling based on trace routing and some methods were studied for eliminating bias of the high degrees in (Ribeiro and Towsley 2010). *Ribeiro et al.* proposed a sampling method, called frontier sampling. It is developed from conventional random walk, which used several dependent random walks. The frontier sampling outperforms conventional random walk and generates small errors in sparse graphs. Sampling for modular structure of networks has been studied by *Maiya et al.* via identifying the communities (Maiya and Berger-Wolf 2010). *Rejaie et al.* tried to estimate the number of users for *MySpace* and *Twitter* by generating sequential user id (Rejaie et al. 2010), but this technique is failed for those online social networks where the user id is randomly generated. Respondent Driven Sampling (RDS) was analyzed in (Gile and Handcock 2010) to reduce the biases associated with chain referral sampling of hidden populations, and then later, sampling from *Twitter* using RDS has been reported to characterize it (Salehi et al. 2011). They have shown through experimentations on *Twitter* that RDS has lower error in comparison with MHRW.

An analytical comparison between RWS and BFS sampling has been presented by *Kurant et al.* (Kurant et al. 2011) for sampling from network. Their study revealed that the degree of graph is overestimated by the BFS, while it is underestimated by the RW sampling. Therefore, they suggested analytical solutions to correct the biasness of estimation. Based on the different types of relationship between users in OSN, multi-graph has been introduced using random walk (Gjoka et al. 2011) and the results of its simulation indicated improvement of the proposed method by *Gjoka.* Random jump in MHRW with prevention of being trapped in local structures has been proposed by *Jin et al.* for

unbiased estimation (Jin et al. 2011). *Pina-Garcia et al.* (Piña-García and Gu 2013) altered the behavior of MHRW using spirals as a probability distribution instead of a classic normal distribution and showed that their algorithm outperforms normal MHRW in case of illusion spiral.

*Lu et al.* studied network sampling on *Twitter* data and their study showed that RW sampling is much better than RN sampling (Lu and Li 2012). Sampling by crawling the edges of graph has been discussed in (Salehi et al. 2012) by the idea of PageRank, which provides more significant results in comparison with RDS. In (Park and Moon 2013) for network sampling, the study has focused on the node attributes of the graph such as estimation of user attribute, user profiles, user tags, user interests, and user preferences of online social networks and then they investigated several network sampling techniques. Directed heterogeneous graph (Yang et al. 2013) is suggested by *yang et al.* for the semantically sampling. They demonstrated that their sampling technique preserves the relational profile property.

Based on the basic snowball sampling a random multiple snowball with *Cohen* process sampling is developed by *Gao et al.* (Gao et al. 2014). Their simulations on computer generated networks indicated that this method is able to preserve local and global structure of the network. In (Luo et al. 2015), two sampling methods are presented by *Peng et al.*, the first sampling method is an improved version of the stratified random sampling method and selects the high degree vertices with higher probability by classifying the nodes according to their degree distribution and the second algorithm is an improved version of snowball sampling method to sample the targeted vertices selectively. Recently, a network sampling method based on distributed learning automata (DLAS) is reported by *Rezvanian et al.* (Rezvanian et al. 2014). It has been shown that this algorithm performs better than RDS and RWS in terms of RE and KS on some well-known real networks. The authors also presented an improved version of this algorithm using extended distributed learning automata (*e*DLA) for sampling online social networks (Rezvanian and Meybodi 2016b) and they showed that due to the ability of *e*DLA in graph traversal, the results of sampled network obtained by the *e*DLA based sampling algorithm is significantly better than that of DLAS. Some concepts of graph theory such as shortest paths (SPS) (Rezvanian and Meybodi 2015b) and spanning trees (SST) (Jalali et al. 2016) are efficiently used for sampling social networks and have shown that their effectiveness by exhaustive simulations on well-known data sets.

In (Wang et al. 2015), due to the high cost of exploring the online social network topology by sampling algorithms, *Wang et al.* presented unbiased sampling methods to characterize user pair properties based on uniform vertex sampling and random walk algorithms. They discovered the significant Homophily in three online social network services (*Foursquare*, *Douban*, and *Xiami*). In (Blagus et al. 2015), the authors studied the presence of characteristic groups of nodes in sampled social and information networks. They considered different network sampling algorithms such as RNS and RES. They observed that the structure of sampled networks exhibits stronger characterization by community-like groups than the original networks, irrespective of their type and consistently across various sampling techniques. In (Tong et al. 2016), the authors proposed an improved version of forest fire sampling algorithm based on PageRank (IFFST-PR). At first IFFST-PR selects marginal nodes between communities of network, then ranks the nodes using PageRank algorithm and finally the sampled network is constructed using the FFS according to the ranked nodes. Similar idea using community structures of network based on two concepts of hierarchical community extraction and densification power law (DPL) was presented by *Yoon et al.* (Yoon et al. 2015), they used hierarchical community extraction to partitions an original network into a set of communities. Then for each community a sampled sub-graph is made by choosing nodes within the community with the probability in proportion to the node degree and ultimately the final sampled

network is constructed by merging sampled sub-graphs based on dendrogram of the hierarchy between a set of communities.

One may also classify the sampling algorithms into two groups: 1) one-phase sampling algorithms which construct the sampled network by random selection of edges/nodes or using some kinds of graph traversal procedure (*e.g.*, RDS (Gile and Handcock 2010), RWS (Yoon et al. 2007), FFS (Kurant et al. 2010), SBS (Frank 2011), to name a few); 2) two-phase sampling algorithms which construct sampled networks using a graph traversal procedure and some pre or post processing (*e.g.*, DLAS (Rezvanian et al. 2014), SPS (Rezvanian and Meybodi 2015b), random PageRank (RPN) (Yoon et al. 2015), DPL (Yoon et al. 2015), IFFST-PR (Tong et al. 2016), to mention a few). The first group of sampling algorithms is relatively simple and has low cost, low accuracy and fails to perform very well on all kinds of networks. The second group uses additional pre or post processing in order to get the information about the network such as classifying important nodes into groups based on Katz centrality measures (Luo et al. 2015), scoring or ordering nodes by PageRank algorithm (Salehi et al. 2012; Tong et al. 2016), extracting groups of nodes in network (Blagus et al. 2015), extracting community structures of network (Yoon et al. 2015) to achieve higher accuracy. Such a pre or post processing certainly increases the cost of the sampling algorithms, which must be paid if achieving higher accuracy is the goal. This should be also noted that the sampled network, once constructed, can be used many times for various applications and analyses.

# 3. Weighted network sampling

Despite the merits of popular network sampling algorithms such as random edge sampling (Leskovec and Faloutsos 2006), random node sampling (Leskovec and Faloutsos 2006), random walk sampling (Yoon et al. 2007), shortest path sampling (Rezvanian and Meybodi 2015b) and distributed learning automata based sampling (Rezvanian et al. 2014) they can only be applied to unweighted networks. In this section, we generalize theses sampling algorithms to be applied to weighted networks. In this section, generalizations of six sampling algorithms to weighted networks are presented among which two algorithms (DLAS (Rezvanian et al. 2014) and SPS (Rezvanian and Meybodi 2015b)) the generalizations of the algorithms are previously proposed by the authors. For all sampling algorithms, we assume that sample ID of nodes is available and weights can take on only positive values.

## 3.1. Weighted random edge sampling (WRES)

Weighted random edge sampling (WRES) algorithm can be generalized for weighted networks by reweighting the probability vector of edge selection for taking sample from each edge. In this respect, each edge is selected with probability $p_{ij}$ defined as follows

$$p_{ij} = \frac{w_{ij}}{\sum_{v_i,v_j \in V} w_{ij}} \tag{4}$$

where $w_{ij}$ is the weight of edge $e_{ij}$ and the denominator calculates the total weights of all edges of the network. Then sampling the selected edge $e_{ij}$ and both adjacent nodes of the selected edge $e_{ij}$ proceeds until the size of the sampled network is reached to a desired size $k=\varphi\times|V|$ where $\varphi$ is the sampling rate. Finally all weights of the sampled edges also select for inclusion in the sampled weight network. The pseudo-code of WRES for weighted network is given in Figure 1.

The computational complexity of WRES is determined by the number of edges of the network $m=|E|$, the number of desired nodes $k$ in sampled network where $k=\varphi\times|V|$. Initialization of WRES for calculating probability vector needs $O(m) \approx O(n^2)$ operations and the process of selection of the edges

needs $O(k)$ operations where $k << n$. Thus, the total computational complexity of WRES is $O(n^2)$ + $O(k) = O(n^2)$.

**Algorithm 1:** WRES($G$, $\varphi$)
**Input:** Weighted network $G = \langle V, E, W \rangle$, Sampling rate $\varphi$
**Output:** Sampled weight network $G' = \langle V', E', W' \rangle$
initial $V' \leftarrow \{\}$, $E' \leftarrow \{\}$
calculate each element of the probability vector $P=\{p_1, \ldots, p_m\}$ according to equation (4)
**while** ($|V'| \leq |V| \times \varphi$) **do**
  select an edge $e_{ij}$ among $E$ at random based on the probability vector $P$
  $V' \leftarrow V' \cup \{v_i, v_j\}$
  $E' \leftarrow E' \cup \{e_{ij}\}$
  **if** ($|V'| > |V| \times \varphi$) **then**
    $V' \leftarrow V' \setminus v_j$
    $E' \leftarrow E' \setminus e_{ij}$
  **end if**
**end while**
$W' \leftarrow \cup_{ij}\{w_{ij} \mid e_{ij} \in E'\}$
**return** $G' = \langle V', E', W' \rangle$

**Figure 1.** Pseudo-code of weighted random edge sampling algorithm.

## 3.2. Weighted random node sampling (WRNS)

We generalize the random node sampling (RNS) algorithm to be applied to weighted networks by reweighting the probability of node selection for taking sample from each node. In this respect, each node is sampled with probability $p_i$ defined as follows

$$p_i = \frac{wd_i}{\sum_{v_i, v_j \in V} w_{ij}} \tag{5}$$

where $wd_i = \sum_{v_j \in N_i} w_{ij}$ is the strength or weight of node $v_i$, $N_i$ is the set of neighbors of node $v_i$ and the denominator of above equation indicates the total weights of all strengths of nodes. Then in the sampling algorithm, the selected node $v_i$ proceeds until the size of the sampled network is reached to a desired size $k = \varphi \times |V|$. Finally all weights and edges among the sampled nodes also select for inclusion in the sampled weight network. The pseudo-code of WRNS for weighted network is given in Figure 2.

The computational complexity of WRNS is determined by the number of nodes of network $n=|V|$, the number of desired nodes $k$, in the sampled network where $k = \varphi \times |V|$. Initialization of WRNS for calculating probability vector of node selection needs $O(n)$ operations and the process of selection of nodes needs $O(k)$ operation where $k << n$. Thus, the total computational complexity of WRNS is $O(n)$ + $O(k) = O(n)$.

**Algorithm 2:** WRNS($G$, $\varphi$)
**Input:** Weighted network $G=\langle V, E, W \rangle$, Sampling rate $\varphi$
**Output:** Sampled weight network $G'=\langle V', E', W' \rangle$
initial $V' \leftarrow \{\}$, $E' \leftarrow \{\}$
calculate each element of the probability vector $P=\{p_1, \ldots, p_n\}$ according to equation (5)
**while** ($|V'| \leq |V| \times \varphi$) **do**
  select a node $v_i$ among $V$ at random based on the probability vector $P$
  $V' \leftarrow V' \cup \{v_i\}$
**end while**

7: $E' \leftarrow \cup_{i,j}\{e_{ij} \mid v_i, v_j \in V'\}$
8: $W' \leftarrow \cup_{ij}\{w_{ij} \mid e_{ij} \in E'\}$
9: **return** $G'=\langle V', E', W' \rangle$

**Figure 2.** Pseudo-code of weighted random node sampling algorithm.

### 3.3. Weighted random walk sampling (WRWS)

The weighted random walk sampling (WRWS) algorithm can be generalized again by reweighting probability of taking sample from each node. The WRWS starts at node $v_i$ randomly and at each iteration selects the next node from one of its neighbors with probability $p_{ij}$ as given below

$$p_{ij} = \frac{w_{ij}}{\sum_{v_j \in N_i} w_{ij}} \tag{6}$$

where $w_{ij}$ is the weight of edge $e_{ij}$, $N_i$ is the set of neighbors of node $v_i$ and the denominator of above equation is the sum of the weights of all neighboring edges. Once the size of sampled nodes of the sampled network reaches to the desired size $k=\varphi \times |V|$, all weights and edges among the sampled nodes also select for inclusion in the sampled weight network. The pseudo-code of WRWS for weighted network is given in Figure 3.

The time needed for WRWS consist of the time needed for calculating probability of selecting next node with $O(D)$ where $D$ ($D << n$) is the degree of that node and the time for walking between selected nodes with $O(1)$ that both process are performed $k$ ($k << n$) times, where $k=\varphi \times |V|$ is the desired size for the sampled network. Thus the total computation complexity for WRWS is $O(n)$.

**Algorithm 3:** WRWS($G$, $\varphi$)
**Input:** Weighted network $G=\langle V, E, W \rangle$, Sampling rate $\varphi$
**Output:** Sampled weight network $G'=\langle V', E', W' \rangle$
1: initial $V' \leftarrow \{\}$, $E' \leftarrow \{\}$
2: select initial node $v_i$ randomly
3: $V' \leftarrow V' \cup \{v_i\}$
4: **while** ($|V'| \leq |V| \times \varphi$) **do**
5: calculate probability $p_{ij}$ according to equation (6)
6: select one of neighboring node $v_i$ with probability $p_{ij}$
7: $V' \leftarrow V' \cup \{v_j\}$
8: $v_i \leftarrow v_j$
9: **end while**
10: $E' \leftarrow \cup_{i,j}\{e_{ij} \mid v_i, v_j \in V'\}$
11: $W' \leftarrow \cup_{ij}\{w_{ij} \mid e_{ij} \in E'\}$
12: **return** $G'=\langle V', E', W' \rangle$

**Figure 3.** Pseudo-code of weighted random walk sampling algorithm.

### 3.4. Weighted metropolis-hasting random walk sampling (WMHRW)

The weighted metropolis-hasting random walk sampling (WMHRW) algorithm starts at node $v_i$ randomly. At each iteration, the algorithm moves from node $v_i$ to its neighbor $v_j$ with probability $p_{ij}$ given below

$$p_{ij} = min\left\{1, \frac{wd_i}{wd_j}\right\} \tag{7}$$

where $wd_i = \sum_{v_k \in N_i} w_{ik}$ is proportional to the sum of all the weights of the edges incident to node $v_i$ and $N_i$ is the set of neighbors of node $v_i$. Then sampling the nodes by moving from node $v_i$ to $v_j$ repeats

until the size of sampled network is reached to the desired size $k = \varphi \times |V|$ where $\varphi$ is the predefined sampling rate. The pseudo-code of WMHRW is given in Figure 4.

The WMHRW algorithm consists of three main steps: 1) generate a random number with time $O(1)$, 2) calculating probability of selecting next node with time $O(D)$ where $D$ ($D << n$) is the degree of that node and 3) moving from current node to the next node with time $O(1)$. Since, these steps are performed sequentially $k$ ($k << n$) times where $k=\varphi \times |V|$ is the desired size for the sampled network, the total computation complexity for WMHRW is $O(n)$.

**Algorithm 4:** WMHRW($G$, $\varphi$)

**Input:** Weighted network $G=\langle V, E, W\rangle$, Sampling rate $\varphi$

**Output:** Sampled weight network $G'=\langle V', E', W'\rangle$

initial $V' \leftarrow \{\}$, $V' \leftarrow \{\}$
select initial node $v_i$ randomly
$V' \leftarrow V' \cup \{v_i\}$
**while** ($|V'| \leq |V| \times \varphi$) **do**
    calculate probability $p_{ij}$ according to equation (7)
    generate random number $r$ uniformly from (0,1) distribution
    **If** $r < p_{ij}$ **then**
        $V' \leftarrow V' \cup \{v_j\}$
        $v_i \leftarrow v_j$;
    **end if**
**end while**
$E' \leftarrow \cup_{ij}\{e_{ij} \mid v_i v_j \in V'\}$
$W' \leftarrow \cup_{ij}\{w_{ij} \mid e_{ij} \in E'\}$
**return** $G'=\langle V', E', W'\rangle$

**Figure 4.** Pseudo-code of weighted metropolis-hastings random walk sampling algorithm.

## 3.5. Weighted distributed learning automata based sampling (WDLAS)

In this section, we give a generalization of the distributed learning automata based sampling (DLAS) (Rezvanian et al. 2014) algorithm for weighted networks. WDLAS uses a set of learning automata in order to guide the process of visiting nodes by visiting the important parts of the input network, those parts of the network which contains more important nodes. A DLA is constructed by assigning a learning automaton $A_i$ to node $v_i$ of the input network. Action-set of each learning automaton corresponds to the edges of the node to which the learning automaton is assigned. Each learning automaton in DLA, initially chooses its actions with equal probabilities. In an iteration of the algorithm, DLA starts from a randomly chosen node, and then follows a sequence of nodes according to the probability vectors of set of learning automata constituting DLA. The sequence of visited nodes is a path with length proportional to $k$ where $k=\varphi \times |V|$ is the desired size for sampled network. The total weigh of visited edges are computed and compared with the total weight of already chosen path in the previous iteration, if this weight is more than the weight of previous one then the probability of actions chosen by all the learning automata along the visited path will be increased according to the learning algorithm. After several iterations, the maximum number of iteration is reached and the visited nodes are sorted according to the number of times that nodes are visited in descending order. The sampled network is then constructed using $k$ mostly visited nodes. Pseudo-code of WDLAS algorithm for sampling weighted networks is given in Figure 5.

The time needed by WDLAS consist of two parts: 1) the time needed for walking in network using DLA with $O(kT)$ where $k=\varphi \times |V|$ is the desired size for sampled network and $T$ ($T \approx n$) is the maximum number of iteration; 2) the time needed for sorting visited nodes with $O(n\log n)$. Thus the total computational complexity for WDLAS is $O(kn) + O(n\log n) = O(n\log n)$.

**Algorithm 5**: WDLAS($G$, $\varphi$, $T$)

**Input:** Weighted network $G=\langle V, E, W\rangle$, Sampling rate $\varphi$, Maximum number of iteration $T$
**Output:** Sampled weight network $G'=\langle V', E', W'\rangle$

initial $V' \leftarrow \{\}$, $E' \leftarrow \{\}$
construct a DLA by assigning an automaton $A_i$ to each node $v_i$
$t$ denotes the iteration number of algorithm which is initially set to 1
$L$ denotes list of visiting node
**while** ($t \leq T$) **do**
    $t \leftarrow t + 1$
    select starting node $v_i$ randomly
    $\pi_k \leftarrow \pi_k \cup \{v_i\}$
    $L(i) \leftarrow L(i) + 1$
    **while** (number of visited nodes $\leq \phi \times |V|$) **do**
        $A_i$ selects an action according to its action probability vector
        let the selected action by $A_s$ be edge ($v_i$ , $v_j$)
        $\pi_k \leftarrow \pi_k \cup \{v_j\}$
        $L(j) \leftarrow L(j) + 1$
        $v_i \leftarrow v_j$
    **end while**
    **if** $W(\pi_k) \geq W(\pi_{k-1})$ **then**
        reward the chosen action by all the learning automata along path $\pi_k$
    **else**
        penalize the chosen action by all the learning automata along path $\pi_k$
    **end If**
**end while**
construct an induced subgraph $G'$ using $\varphi \times |V|$ mostly visited nodes using list $L$
**return** $G'=\langle V', E', W'\rangle$

**Figure 5.** Pseudo-code of weighted distributed learning automata based sampling.

## 3.6. Weighted shortest path sampling (WSPS)

In this section, we generalize the shortest path sampling (SPS) (Rezvanian and Meybodi 2015b) algorithm for weighted networks. The WSPS algorithm iteratively computes shortest paths between pairs of nodes for $T$ times. At iteration $t$, WSPS chooses randomly two non-adjacent nodes $v_s$ as source node and $v_d$ *as* destination node and then computes the shortest path $\pi_k$ between these nodes using a shortest path algorithm such as *Dijkstra's* algorithm (Dijkstra 1959). Then, the visited nodes are sorted according to the number of shortest paths along which that node has been appeared. The sampled network now can be constructed by considering a sub-graph of the input network whose node-set contains $k=\varphi \times |V|$ mostly visited nodes. The pseudo-code of WSPS for sampling weighted networks is given in Figure 6.

The time needed by WSPS consists of two parts: 1) the time needed for computing the shortest paths and 2) the time needed to sort the visited nodes appearing in the computed shortest paths. Since the number of computed shortest paths is a percentage of the number of nodes in the graph ($T << n$) and the computation of a shortest path is proportional to $O(n^2)$ then the time required by the first part is $O(Tn^2) \approx O(n^2)$. Sorting the nodes appearing in the computed shortest paths in the worst case takes $O(n\log n)$ and hence the computational complexity of WSPS is $O(n^2) + O(n\log n) = O(n^2)$.

**Algorithm 6**: WSPS($G$, $\varphi$, $T$)

**Input:** Weighted network $G=\langle V, E, W\rangle$, Sampling rate $\varphi$, Maximum number of computed shortest paths $T$.
**Output:** Sampled weight network $G'=\langle V', E', W'\rangle$

initial $V' \leftarrow \{\}$, $E' \leftarrow \{\}$
$t$ denotes the iteration number of algorithm which is initially set to 1
$\pi_t$ denotes the shortest path between $v_s$ and $v_d$ at iteration $t$
$L$ denotes list of visiting node

5: **while** ($t < T$) **do**
6: select two non-adjacent source nodes $v_s$ and destination $v_d$ randomly
7: compute the shortest path $\pi_t$ between $v_s$ and $v_d$
8: $L(\pi_t) \leftarrow L(\pi_t) + 1$
9: $t \leftarrow t + 1$
10: **end while**
11: construct an induced subgraph $G'$ using $\phi \times |V|$ mostly visited nodes using list $L$
12: **return** $G' = \langle V', E', W' \rangle$

**Figure 6.** The pseudo-code of the weighted shortest path sampling algorithm.

# 4. Simulation Results

In this section, performance of the proposed weighted sampling algorithms is investigated on several well-known real weighted networks. Table 1 describes the characteristics of real networks used for the experimentations. These networks are 2010 US airport network (Opsahl et al. 2010), Facebook-like social network (Opsahl and Panzarasa 2009), network of computational geometry collaborations (Beebe 2002), non-US open flights airports (Opsahl et al. 2010), networks of co-authorships (Newman 2001), collaboration network of Arxiv Astro Physics (Leskovec et al. 2007), co-authorship network of scientists (Newman 2006), Internet routers (2016a), network of US patents (Hall et al. 2001) and network of English words (2016b).

**Table 1.** Description of test weighted networks.

| Network | Node | Edge | Type | Description |
|---|---|---|---|---|
| *2010 US airport network* (Opsahl et al. 2010) | 1574 | 28236 | Directed | Network of the complete US airport network in 2010. The weights correspond to the number of seats available on the scheduled flights. |
| *Facebook-like Social Network* (Opsahl and Panzarasa 2009) | 1899 | 20296 | Directed | Network of an online community for students at University of California, Irvine, based on exchange at least one message among users and each edge weighted by number of messages. |
| *Geom* (Beebe 2002) | 7343 | 23796 | Undirected | Network of Computational Geometry collaborations based on common publications between two authors. Each edge weighted by the number of common publications between two authors |
| *Openflights* (Opsahl et al. 2010) | 7976 | 30501 | Directed | Network of two non-US-based airports. The weights in this network refer to the number of routes between two airports. |
| *Cond-mat-1999* (Newman 2001) | 16726 | 47594 | Undirected | Networks of co-authorships for scientific preprints posted to the condensed matter archive (cond-mat), based on submissions beginning in 1995 and continuing through 1999. The weight of each edge is sum of joint papers. |
| *Astro-ph* (Leskovec et al. 2007) | 18772 | 198110 | Undirected | Collaboration network of Arxiv Astro Physics. Weight of each edge is the number of common co-authored papers. |
| *Netscience* (Newman 2006) | 1589 | 5484 | Undirected | Co-authorship network of scientists working on network theory and experiments. The weights assigned directly in terms of the number of collaborations between authors and inversely in terms of the number of |

| | | | | |
|---|---|---|---|---|
| | | | | other authors involved. |
| *Internet routers* (2016a) | 124651 | 207214 | Directed | The main core of subgraph extracted from network of connectivity of internet routers. The weight of each edge is the number of routes between internet routers. |
| *Pajek/patents_main* (Hall et al. 2001) | 240547 | 560943 | Directed | Network of containing information on almost 3 million U.S. patents granted between January 1963 and December 1999, the weight associated to each edge is the citations made to these patents between 1975 and 1999. |
| *Pajek/Wordnet3* (2016b) | 82670 | 132964 | Directed | Network of English words as nodes and various relationships between them as edges. The weight of each edge is proportional to the number of any relationships exist between pair of nodes. |

## 4.1. Distance measures

Distance between a property of original networks and that of sampled network is often calculated for evaluating the quality of sampled network. In this paper, we use the Kolmogorov-Smirnov (KS) statistic, skew divergence (SD), Pearson's correlation coefficient (PCC) and relative error (RE) as distance measures to compare different weighted sampling algorithms proposed in this paper. These distance measures as evaluating criteria are described below.

### 4.1.1. Kolmogorov-Smirnov (KS) Statistic

*Kolmogorov-Smirnov* (KS) statistic is one of the statistical test methods commonly used for assessment the distance between two cumulative distribution functions (CDFs). The KS measures acceptability between original distribution and estimated distribution. The result of this test is a value between 0 and 1 that as closer as it is to zero, both distributions will have a greater similarity; and as closer as it is to unit, the two distributions will show a greater discrepancy. This measure has been defined as

$$KS(P,Q) = \underset{x}{\text{MAX}}|P(x) - Q(x)| \tag{8}$$

where $P$ and $Q$ are two CDFs of original and estimated data, respectively and $x$ represents the range of the random variables. So it is computed as the maximum vertical distance between the two distributions. The result of this test can be employed for comparison of sampling methods (Jalali et al. 2015).

### 4.1.2. Skew Divergence (SD)

Skew divergence (Jalali et al. 2016) can also be used for assessment the distance between two probability distribution functions (PDFs) and defined as follows

$$SD(P,Q,\alpha) = D[\alpha P + (1-\alpha)Q||\alpha Q + (1-\alpha)P] \tag{9}$$

where $D$ is the *Kullback-Leibler* (KL) *divergence*, which measures the similarity between two PDFs $P$ and $Q$ that do not have continuous support over the full range of values and $\alpha$=0.99. The KL *divergence* is defined as follows

$$KL(P||Q) = \sum_x P(x) log \frac{P(x)}{Q(x)} \tag{10}$$

### 4.1.3. Pearson's Correlation Coefficient (PCC)

One of the good indices to measure the similarity between the sampled parameter values and the real parameters values is Pearson's correlation coefficient (PCC) and calculated as follows

$$PCC(P,Q) = \frac{n\sum_{i=1}^{n} p_i q_i - (\sum_{i=1}^{n} p_i)(\sum_{i=1}^{n} q_i)}{\sqrt{n\sum_{i=1}^{n} p_i^2 - (\sum_{i=1}^{n} p_i)^2}\sqrt{n\sum_{i=1}^{n} q_i^2 - (\sum_{i=1}^{n} q_i)^2}} \tag{11}$$

where $x_i$ and $x_i'$ are the values of the real and sampled parameters (*i.e.*, real Eigen values and sampled Eigen values) in the original data and sampled data respectively, $n$ is the number of sampled data. If the result of PCC is much closer to 1, then the performance of a sampling method is much better (Rezvanian and Meybodi 2016a).

### 4.1.4. Relative Error (RE)

Relative error (RE) can be applied to assess accuracy of the results for a single parameter, which is defined by the following equation

$$RE = \frac{|P - Q|}{P} \tag{12}$$

where, $P$ and Q denote the values of real and sampled parameters (*i.e.*, real clustering coefficient and estimated clustering coefficient) in the original data and sampled data respectively (Rezvanian and Meybodi 2015b).

## 4.2. Experimental Results

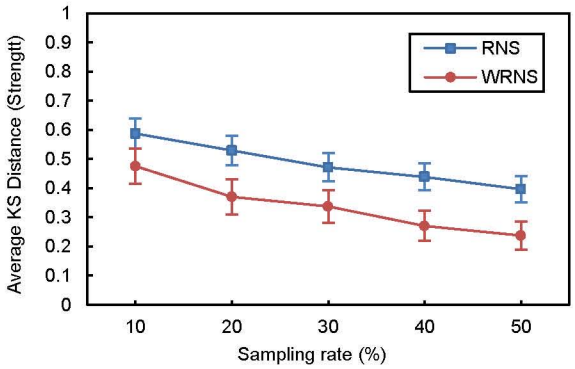
To study the performance of the proposed weighted sampling algorithms, several experiments are conducted on several weighted real networks as described in Table 1. Note that the proposed weighted sampling algorithms are abbreviated as follows: Weighted Random Node Sampling as WRNS, Weighted Random Edge Sampling as WRES, Weighted Random Walk Sampling as WRWS, Weighted Metropolis-hasting Random Walk sampling as WMHRW, Weighted distributed learning automata based sampling as WDLAS and Weighted shortest path sampling as WSPS. For WDLAS, learning rate is set to 0.05, and the maximum number of iteration $T$ is set to $n\times100$, where $n$ is the number of nodes in the given network. For WSPS, number of computed shortest paths $T$ is set $\varphi\times n$, where $n$ is the number of nodes of the network and $\varphi$ is the sampling rate. For all experiments, the sampling rate is varied from 10 to 50% with 10% interval and the results over all test networks are reported.

### 4.2.1. Experiment I

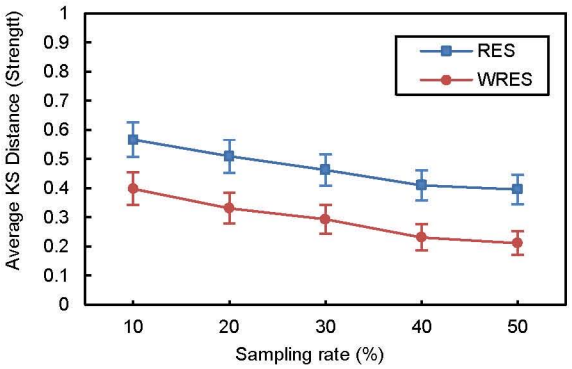
This experiment is conducted to study the performance of the proposed weighted sampling algorithms for weighted networks versus when the existing unweighted sampling algorithms have been used in terms of the KS distance for weighted degree (strength) distribution and KS distance for weighted clustering coefficient distribution. For this purpose, the sampling algorithms are performed on the test networks for varying sampling rate from 10 to 50% with 10% interval. The results of this experiment which are given in Figure 7 in terms of KS distance for strength distribution and figure 8 in terms of

KS distance for clustering coefficient distribution. In Figure 7 and Figure 8, the points along the curves show the average and standard deviation value over all the test networks. As it is shown, all algorithms have better performance when the sampling rate is higher and the proposed weighted sampling algorithms perform better than their unweighted versions, which means that the proposed sampling algorithms for weighted networks can sample more appropriate nodes and edges than their unweighted versions, therefore, the weighted sampling algorithms tend to have a better performance on the aspect of keeping structure of the original weighted network.

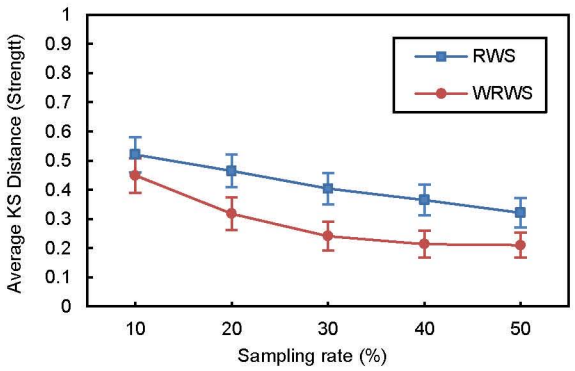
Moreover, in order to investigate the significance of the weighted sampling algorithms for the weighted networks versus when existing unweighted sampling algorithms has been used, we provided *t*-Test to compare the results of this comparison. In this test, the statistical results for comparing the weighted sampling algorithms with their unweighted versions by the two-tailed *t*-Test with 28 degrees of freedom at a 0.05 level of significance are reported for results of average and standard deviation of KS distance for strength and clustering coefficient distributions. In this test, it is assumed that the difference between each pair of algorithm is statistically significant, if the difference significance is smaller than 0.05. The first column of these tables includes the list of the proposed weighted sampling algorithms together with the unweighted sampling algorithms. The second and third columns show the average KS distance ± standard deviation for each sampling algorithms. The second column shows that WSPS has the best results and WRNS has the worst results among all weighted sampling algorithms. Symbols "+" and "-" appeared in the column labeled as "Performance" indicates that the performance of each weighted sampling algorithm is significantly better than or worse than the corresponding unweighted sampling algorithm, respectively. This conclusion is drawn on the basis of the average KS distance for strength distribution and difference significance between two comparing algorithms. For example in table 2, for comparison between WSPS and SPS difference significance of 3.89E-04 show that WSPS outperforms SPS. For each pair comparison in table 2 and table 3, the best result is highlighted in boldface. Based on the test results, the weighted sampling algorithms outperform their unweighted versions. In addition, one can conclude that in general, the edge weight of the weighted networks as a natural component of the networks is important and the process of sampling by weighted sampling algorithms, in which the edge weight of the networks are considered, leads to significantly better sampling compared when the unweighted sampling algorithms are used.



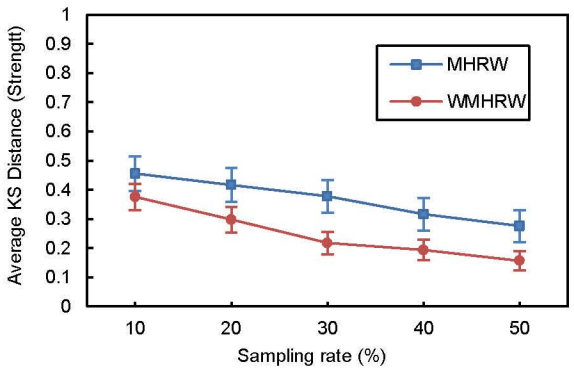

**a)** RNS vs. WRNS



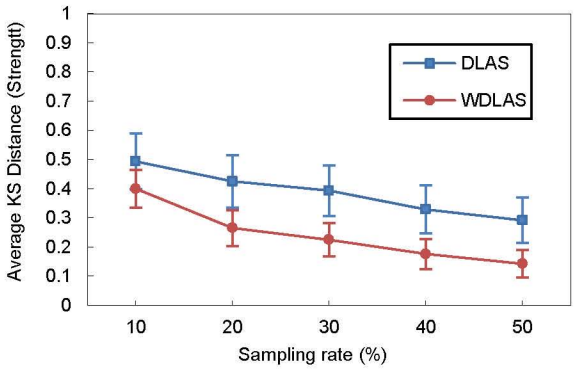

**b)** RES vs. WRES



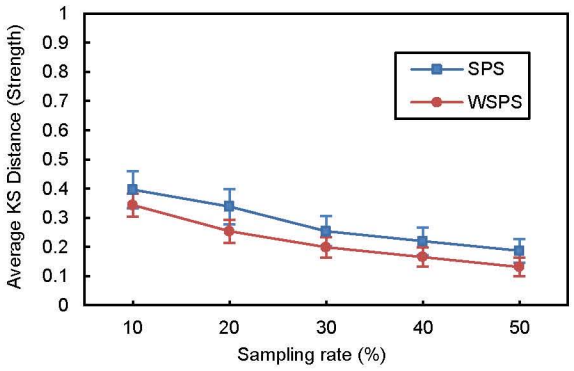

**c)** RWS vs. WRWS




**d)** MHRW vs. WMHRW

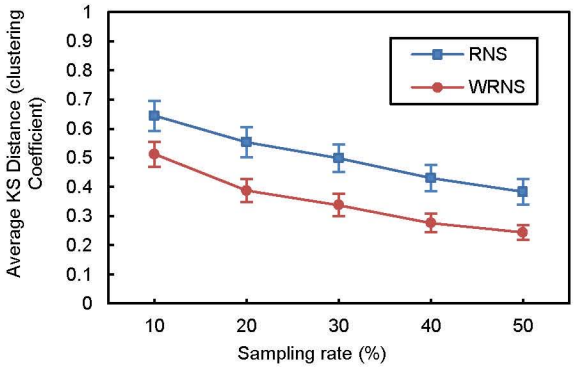



**e)** DLAS vs. WDLAS

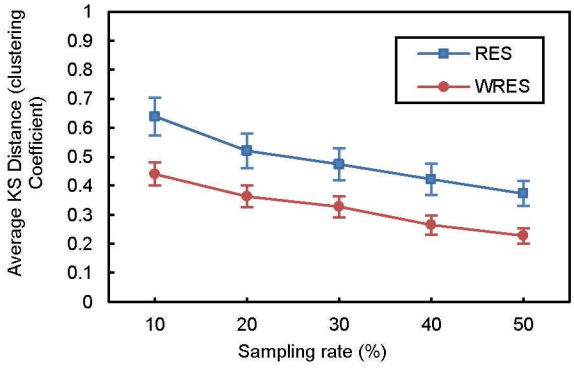



**f)** SPS vs. WSPS

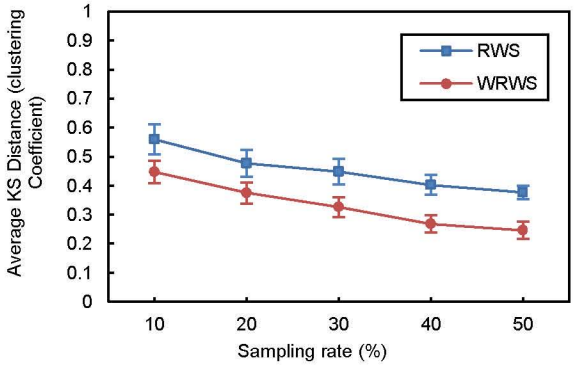
**Figure 7.** Comparing weighted sampling algorithms with their unweighted versions in terms of KS distance for strength distributions.



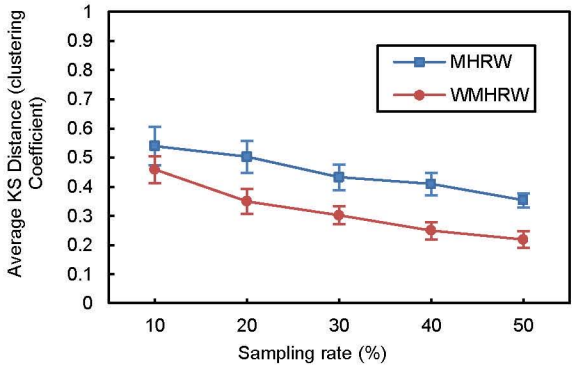

**a)** RNS vs. WRNS



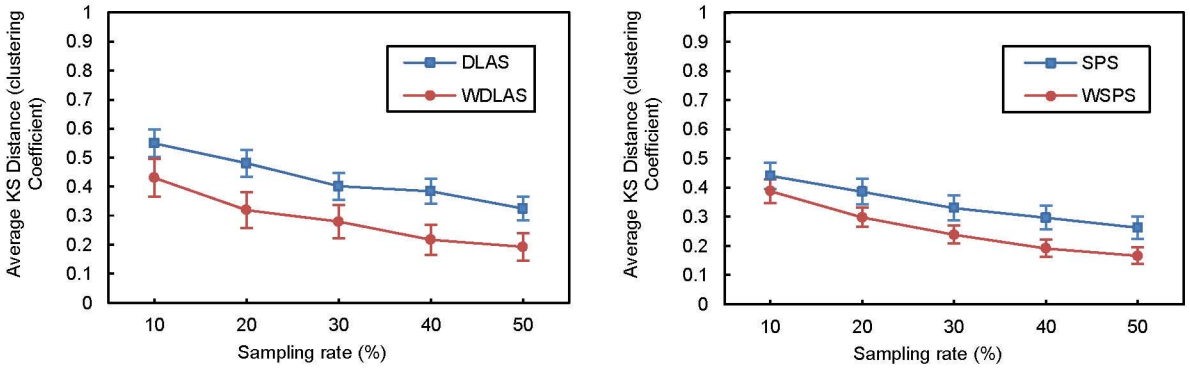

**b)** RES vs. WRES




**c)** RWS vs. WRWS




**d)** MHRW vs. WMHRW




**e)** DLAS vs. WDLAS **f)** SPS vs. WSPS

**Figure 8.** Comparing weighted sampling algorithms with their unweighted versions in terms of KS distance for clustering coefficient distribution.

**Table 2.** Comparing statistical significance of weighted sampling algorithms with their unweighted versions in terms of average KS distance and standard deviation for strength distribution.

| Sampling algorithms | Average results [± standard deviation (std)] | | Different significance | Performance |
|---|---|---|---|---|
| | Unweighted | Weighted | | |
| RNS vs. WRNS | 0.484 ± 0.075 | **0.338 ± 0.057** | 7.16E-09 | + |
| RES vs. WRES | 0.469 ± 0.071 | **0.293 ± 0.058** | 7.79E-11 | + |
| RWS vs. WRWS | 0.419 ± 0.079 | **0.286 ± 0.050** | 4.49E-08 | + |
| MHRW vs. WMHRW | 0.368 ± 0.073 | **0.248 ± 0.044** | 4.96E-08 | + |
| DLAS vs. WDLAS | 0.386 ± 0.078 | **0.241 ± 0.045** | 3.63E-09 | + |
| SPS vs. WSPS | 0.279 ± 0.067 | **0.218 ± 0.038** | 3.89E-04 | + |

**Table 3.** Comparing statistical significance of weighted sampling algorithms with their unweighted versions in terms of average KS distance and standard deviation for weighted clustering coefficient distribution.

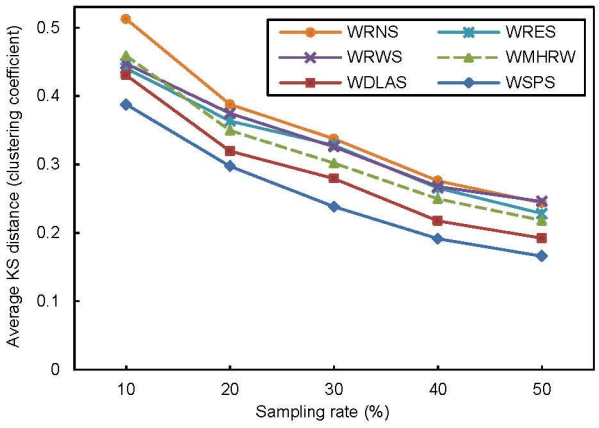
| Sampling algorithms | Average results [± standard deviation (std)] | | Different significance | Performance |
|---|---|---|---|---|
| | Unweighted | Weighted | | |
| RNS vs. WRNS | 0.502 ± 0.083 | **0.351 ± 0.062** | 3.07E-08 | + |
| RES vs. WRES | 0.486 ± 0.082 | **0.325 ± 0.061** | 5.35E-09 | + |
| RWS vs. WRWS | 0.452 ± 0.058 | **0.332 ± 0.056** | 1.77E-08 | + |
| MHRW vs. WMHRW | 0.448 ± 0.053 | **0.316 ± 0.052** | 4.03E-10 | + |
| DLAS vs. WDLAS | 0.428 ± 0.057 | **0.288 ± 0.054** | 3.75E-10 | + |
| SPS vs. WSPS | 0.343 ± 0.048 | **0.256 ± 0.047** | 2.51E-07 | + |

### 4.2.2. Experiment II

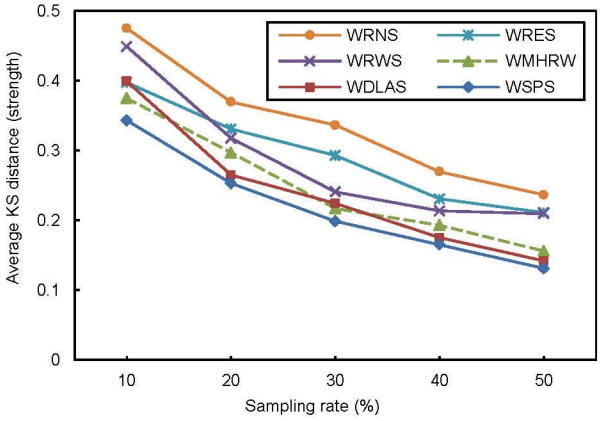
This experiment is carried out to study the performance of the proposed weighted sampling algorithms in terms of KS distance and skew divergence for clustering coefficient and strength distributions and also PCC for Eigen values. For this purpose, the sampling algorithms are performed on the test networks for varying sampling rate from 10 to 50% with 10% interval and the average results over all the test networks are presented. Figure 9.a) and Figure 9.b) show the results of KS distance for average KS distance for clustering coefficient distribution and average KS distance for strength distribution, respectively, Figure 9.c) and Figure 9.d) show the results of average skew divergence for clustering coefficient distribution and average skew divergence for strength

distribution, respectively, and Figure 9.e) and Figure 9.f) show the results of average relative error for weighted clustering coefficient and average PCC for Eigen values, respectively.

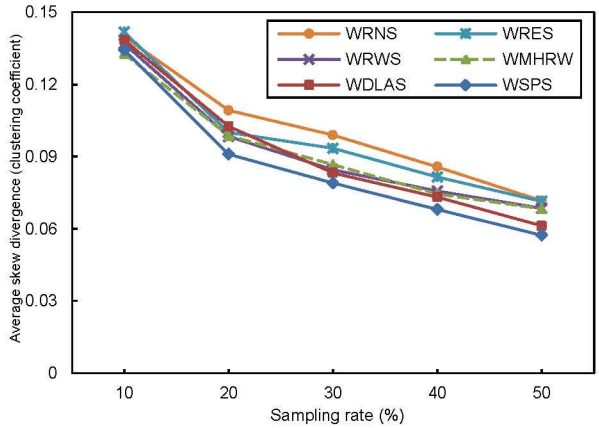
According to the results, one may be concluded that for all the test networks, increasing the sampling rate results in increasing PCC for Eigen values, decreasing KS distance and skew divergence for both clustering coefficient and strength distributions, which means all sampling algorithms will leads to a better performance on the aspect of maintaining the structure of the original network when the sampling rate increases. This can be due to the fact that network sampling is generally harder for lower sampling rate regardless of the algorithm. From the results given in Figure 9.a) and Figure 9.b) we also observe that among the proposed weighted sampling algorithms, WSPS outperforms other weighted sampling algorithms for different sampling rates in terms of average KS distance for clustering coefficient and average KS distance for strength. From the results given in Figure 9.f), one can be said that the WSPS and WRNS have the highest and the lowest performance, respectively, in terms of average PCC for Eigen values. For other metrics, the results are almost similar in terms of average skew divergence for clustering coefficient, average skew divergence for strength distribution and average PCC for Eigen values.



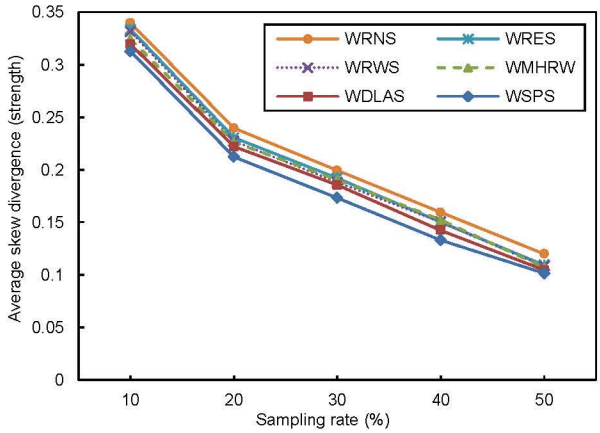

**a)** Average KS distance for clustering coefficient distribution



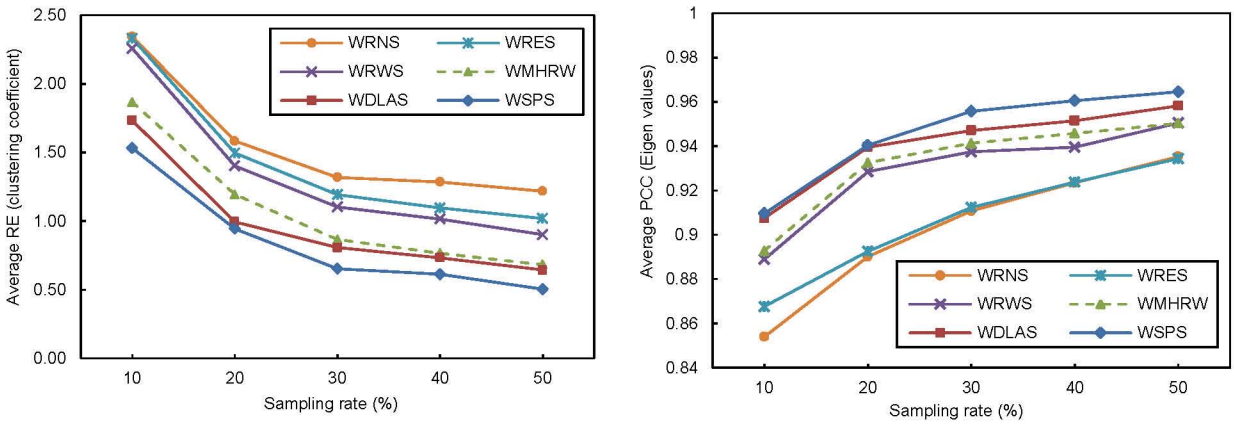

**b)** Average KS distance for strength distribution




**c)** Average Skew divergence for clustering coefficient distribution




**d)** Average Skew divergence for strength distribution




**e)** Average relative error for weighted clustering coefficient

**f)** Average PCC for Eigen values

**Figure 9.** Comparing proposed weighted sampling algorithms in terms of distance measures.

Moreover, To investigate whether a statistical significant difference exists among all the proposed sampling algorithms, we conducted a series of multi-comparison statistical tests (Friedman and Iman-Davenport) with a significance interval of 95% ($\alpha$ = 0.05) (García et al. 2009). Friedman test (Friedman 1940) is used as a non-parametric test that ranks the weighted sampling algorithms with respect to all mentioned test metrics. The rank 1, will be assigned to the best algorithm, the second best, rank 2, and so on. As a statistical analysis, Friedman's test was first applied to obtain rankings. To obtain the adjusted p-values for each comparison between the control algorithm (the best-performing one) and the other algorithms, Holm and Hochberg tests were conducted as post hoc methods (if significant differences are detected).

The rankings obtained by Friedman's test in terms of KS distance for clustering coefficient and strength distributions, skew divergence for clustering coefficient and strength distributions, relative error for weighted clustering coefficient and PCC for Eigen values are given in Table 4. According to the results of average rankings based on statistical significance in Table 4, one can conclude that WSPS and WRNS are ranked the first and the last, respectively. The p-values computed by the Friedman's test for KS distance for clustering coefficient and strength distributions, skew divergence for clustering coefficient and strength distributions, relative error for weighted clustering coefficient and PCC for Eigen values are 4.23E-4, 4.02E-4, 1.54E-3, 2.69E-3, 1.39E-4 and 2.09E-4 respectively which are below the significance interval of 95% ($\alpha$=0.05) for all test metrics. Thus, a significant difference exists among the observed results. Post hoc methods (*Holm* and *Shaffer* tests) are also performed to obtain the adjusted p-values. The adjusted *p*-values of the *Holm* and *Shaffer* tests are given in tables 5-10 for all test metrics. In this test, the null hypothesis is that all the weighted sampling algorithms are equivalent, if the null-hypothesis is rejected, we can compare all the weighted sampling algorithms with each other using the *Nemenyi* test (Nemenyi 1962). The results of adjusted *p*-values in tables 5-10 also indicate that WSPS outperforms the other weighted sampling algorithms (WRNS, WRES, WRWS, WMHRW, WDLAS, WSPS) with significance interval of 95% ($\alpha$=0.05).

**Table 4.** Average rankings of Friedman's test of the comparing weighted sampling algorithms (WRNS, WRES, WRWS, WMHRW, WDLAS and WSPS) with respect to on test metrics.

| **Test metrics** | **WRNS** | **WRES** | **WRWS** | **WMHRW** | **WDLAS** | **WSPS** |
|---|---|---|---|---|---|---|
| KS distance for clustering coefficient distribution | 5.8 | 4.0 | 4.8 | 3.4 | 2.0 | 1.0 |
| KS distance for strength distribution | 6.0 | 4.6 | 4.2 | 2.6 | 2.6 | 1.0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Skew divergence for clustering coefficient distribution | 5.8 | 5.0 | 3.2 | 2.8 | 3 | 1.2 |
| Skew divergence for strength distribution | 6.0 | 4.8 | 3.6 | 3.6 | 2.0 | 1.0 |
| Relative error for weighted clustering coefficient | 6.0 | 5.0 | 4.0 | 3.0 | 2.0 | 1.0 |
| PCC for Eigen values | 5.8 | 5.2 | 3.8 | 3.2 | 2.0 | 1.0 |

**Table 5**. *p*-values of post hoc comparisons with $\alpha$=0.05 for the comparing weighted sampling algorithms (WRNS, WRES, WRWS, WMHRW, WDLAS and WSPS) based on KS distance for clustering coefficient distribution.

| *i* | Weighted sampling algorithms | $z=(R_0 - R_i)/SE$ | *p*-value | Holm *p*-value | Shaffer *p*-value |
|---|---|---|---|---|---|
| 1 | WRNS vs. WSPS | 4.0567 | 0.0001 | 0.0033 | 0.0033 |
| 2 | WRNS vs. WDLAS | 3.2116 | 0.0013 | 0.0036 | 0.0050 |
| 3 | WRWS vs. WSPS | 3.2116 | 0.0013 | 0.0038 | 0.0050 |
| 4 | WRES vs. WSPS | 2.5355 | 0.0112 | 0.0042 | 0.0050 |
| 5 | WRWS vs. WDLAS | 2.3664 | 0.0180 | 0.0045 | 0.0050 |
| 6 | WMHRW vs. WSPS | 2.0284 | 0.0425 | 0.0050 | 0.0050 |
| 7 | WRNS vs. WMHRW | 2.0284 | 0.0425 | 0.0056 | 0.0056 |
| 8 | WRES vs. WDLAS | 1.6903 | 0.0910 | 0.0063 | 0.0063 |
| 9 | WRNS vs. WRES | 1.5213 | 0.1282 | 0.0071 | 0.0071 |
| 10 | WMHRW vs. WDLAS | 1.1832 | 0.2367 | 0.0083 | 0.0083 |
| 11 | WRWS vs. WMHRW | 1.1832 | 0.2367 | 0.0100 | 0.0100 |
| 12 | WRNS vs. WRWS | 0.8452 | 0.3980 | 0.0125 | 0.0125 |
| 13 | WDLAS vs. WSPS | 0.8452 | 0.3980 | 0.0167 | 0.0167 |
| 14 | WRES vs. WRWS | 0.6761 | 0.4990 | 0.0250 | 0.0250 |
| 15 | WRES vs. WMHRW | 0.5071 | 0.6121 | 0.0500 | 0.0500 |

**Note:** Nemenyi's procedure rejects those hypotheses that have a *p*-value ≤ 0.0033; Holm's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0041; Shaffer's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0033.

**Table 6.** *p*-values of post hoc comparisons with $\alpha$=0.05 for the comparing weighted sampling algorithms (WRNS, WRES, WRWS, WMHRW, WDLAS and WSPS) based on KS distance for strength distribution.

| *i* | Weighted sampling algorithms | $z=(R_0 - R_i)/SE$ | *p*-value | Holm *p*-value | Shaffer *p*-value |
|---|---|---|---|---|---|
| 1 | WRNS vs .WSPS | 0.0000 | 0.0004 | 0.0004 | 0.0004 |
| 2 | WRNS vs .WDLAS | 0.0007 | 0.0108 | 0.0101 | 0.0072 |
| 3 | WRES vs .WSPS | 0.0013 | 0.0198 | 0.0172 | 0.0132 |
| 4 | WRES vs .WDLAS | 0.0180 | 0.2694 | 0.2155 | 0.1796 |
| 5 | WRWS vs .WSPS | 0.0280 | 0.4199 | 0.3079 | 0.2799 |
| 6 | WMHRW vs .WSPS | 0.0280 | 0.4199 | 0.3079 | 0.2799 |
| 7 | WRNS vs .WMHRW | 0.0425 | 0.6378 | 0.3827 | 0.2977 |
| 8 | WRNS vs .WRWS | 0.0425 | 0.6378 | 0.3827 | 0.2977 |
| 9 | WRWS vs .WDLAS | 0.1763 | 2.6444 | 1.2341 | 1.2341 |
| 10 | WMHRW vs .WDLAS | 0.1763 | 2.6444 | 1.2341 | 1.2341 |
| 11 | WRNS vs .WRES | 0.3105 | 4.6574 | 1.5525 | 1.2420 |
| 12 | WRES vs .WMHRW | 0.3105 | 4.6574 | 1.5525 | 1.2420 |
| 13 | WRES vs .WRWS | 0.3105 | 4.6574 | 1.5525 | 1.2420 |
| 14 | WDLAS vs .WSPS | 0.3980 | 5.9704 | 1.5525 | 1.2420 |
| 15 | WRWS vs .WMHRW | 1.0000 | 15.0000 | 1.5525 | 1.2420 |

**Note:** Nemenyi's procedure rejects those hypotheses that have a *p*-value ≤ 0.0033; Holm's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0038; Shaffer's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0033.

**Table 7.** *p*-values of post hoc comparisons with $\alpha$=0.05 for the comparing weighted sampling algorithms (WRNS, WRES, WRWS, WMHRW, WDLAS and WSPS) based on skew divergence for clustering coefficient distribution.

| *i* | Weighted sampling algorithms | $z=(R_0 - R_i)/SE$ | *p*-value | Holm *p*-value | Shaffer *p*-value |
|---|---|---|---|---|---|
| 1 | WRNS vs. WSPS | 3.8877 | 0.0001 | 0.0033 | 0.0033 |
| 2 | WRES vs. WSPS | 3.2116 | 0.0013 | 0.0036 | 0.0050 |
| 3 | WRNS vs. WMHRW | 2.5355 | 0.0112 | 0.0038 | 0.0050 |
| 4 | WRNS vs. WDLAS | 2.3664 | 0.0180 | 0.0042 | 0.0050 |
| 5 | WRNS vs. WRWS | 2.1974 | 0.0280 | 0.0045 | 0.0050 |

| | | | | | |
|---|---|---|---|---|---|
| 6 | WRES vs. WMHRW | 1.8593 | 0.0630 | 0.0050 | 0.0050 |
| 7 | WRES vs. WDLAS | 1.6903 | 0.0910 | 0.0056 | 0.0056 |
| 8 | WRWS vs. WSPS | 1.6903 | 0.0910 | 0.0063 | 0.0063 |
| 9 | WDLAS vs. WSPS | 1.5213 | 0.1282 | 0.0071 | 0.0071 |
| 10 | WRES vs. WRWS | 1.5213 | 0.1282 | 0.0083 | 0.0083 |
| 11 | WMHRW vs. WSPS | 1.3522 | 0.1763 | 0.0100 | 0.0100 |
| 12 | WRNS vs. WRES | 0.6761 | 0.4990 | 0.0125 | 0.0125 |
| 13 | WRWS vs. WMHRW | 0.3381 | 0.7353 | 0.0167 | 0.0167 |
| 14 | WRWS vs. WDLAS | 0.1690 | 0.8658 | 0.0250 | 0.0250 |
| 15 | WMHRW vs. WDLAS | 0.1690 | 0.8658 | 0.0500 | 0.0500 |

**Note:** Nemenyi's procedure rejects those hypotheses that have a $p$-value ≤ 0.0033; Holm's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0038; Shaffer's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0033.

**Table 8.** $p$-values of post hoc comparisons with $\alpha$=0.05 for the comparing weighted sampling algorithms (WRNS, WRES, WRWS, WMHRW, WDLAS and WSPS) based on skew divergence for strength distribution.

| $i$ | Weighted sampling algorithms | $z=(R_0 - R_i)/SE$ | $p$-value | Holm $p$-value | Shaffer $p$-value |
|---|---|---|---|---|---|
| 1 | WRNS vs. WSPS | 4.2258 | 0.0000 | 0.0033 | 0.0033 |
| 2 | WRES vs. WSPS | 3.0426 | 0.0023 | 0.0036 | 0.0050 |
| 3 | WRNS vs. WMHRW | 2.8735 | 0.0041 | 0.0038 | 0.0050 |
| 4 | WRNS vs. WDLAS | 2.8735 | 0.0041 | 0.0042 | 0.0050 |
| 5 | WRWS vs. WSPS | 2.7045 | 0.0068 | 0.0045 | 0.0050 |
| 6 | WRES vs. WMHRW | 1.6903 | 0.0910 | 0.0050 | 0.0050 |
| 7 | WRES vs. WDLAS | 1.6903 | 0.0910 | 0.0056 | 0.0056 |
| 8 | WRNS vs. WRWS | 1.5213 | 0.1282 | 0.0063 | 0.0063 |
| 9 | WRWS vs. WMHRW | 1.3522 | 0.1763 | 0.0071 | 0.0071 |
| 10 | WRWS vs. WDLAS | 1.3522 | 0.1763 | 0.0083 | 0.0083 |
| 11 | WMHRW vs. WSPS | 1.3522 | 0.1763 | 0.0100 | 0.0100 |
| 12 | WDLAS vs. WSPS | 1.3522 | 0.1763 | 0.0125 | 0.0125 |
| 13 | WRNS vs. WRES | 1.1832 | 0.2367 | 0.0167 | 0.0167 |
| 14 | WRES vs. WRWS | 0.3381 | 0.7353 | 0.0250 | 0.0250 |
| 15 | WMHRW vs. WDLAS | 0.0000 | 1.0000 | 0.0500 | 0.0500 |

**Note:** Nemenyi's procedure rejects those hypotheses that have a $p$-value ≤ 0.0033; Holm's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0041; Shaffer's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0033.

**Table 9.** $p$-values of post hoc comparisons with $\alpha$=0.05 for the comparing weighted sampling algorithms (WRNS, WRES, WRWS, WMHRW, WDLAS and WSPS) based on relative error for weighted clustering coefficient distribution.

| $i$ | Weighted sampling algorithms | $z=(R_0 - R_i)/SE$ | $p$-value | Holm $p$-value | Shaffer $p$-value |
|---|---|---|---|---|---|
| 1 | WRNS vs. WSPS | 4.2258 | 0.0000 | 0.0033 | 0.0033 |
| 2 | WRNS vs. WDLAS | 3.3806 | 0.0007 | 0.0036 | 0.0050 |
| 3 | WRES vs. WSPS | 3.3806 | 0.0007 | 0.0038 | 0.0050 |
| 4 | WRNS vs. WMHRW | 2.5355 | 0.0112 | 0.0042 | 0.0050 |
| 5 | WRES vs. WDLAS | 2.5355 | 0.0112 | 0.0045 | 0.0050 |
| 6 | WRWS vs. WSPS | 2.5355 | 0.0112 | 0.0050 | 0.0050 |
| 7 | WRNS vs. WRWS | 1.6903 | 0.0910 | 0.0056 | 0.0056 |
| 8 | WRES vs. WMHRW | 1.6903 | 0.0910 | 0.0063 | 0.0063 |
| 9 | WRWS vs. WDLAS | 1.6903 | 0.0910 | 0.0071 | 0.0071 |
| 10 | WMHRW vs. WSPS | 1.6903 | 0.0910 | 0.0083 | 0.0083 |
| 11 | WRNS vs. WRES | 0.8452 | 0.3980 | 0.0100 | 0.0100 |
| 12 | WRES vs. WRWS | 0.8452 | 0.3980 | 0.0125 | 0.0125 |
| 13 | WRWS vs. WMHRW | 0.8452 | 0.3980 | 0.0167 | 0.0167 |
| 14 | WMHRW vs. WDLAS | 0.8452 | 0.3980 | 0.0250 | 0.0250 |
| 15 | WDLAS vs. WSPS | 0.8452 | 0.3980 | 0.0500 | 0.0500 |

**Note:** Nemenyi's procedure rejects those hypotheses that have a $p$-value ≤ 0.0033; Holm's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0041; Shaffer's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0033.

**Table 10.** *p*-values of post hoc comparisons with $\alpha$=0.05 for the comparing weighted sampling algorithms (WRNS, WRES, WRWS, WMHRW, WDLAS and WSPS) based PCC for Eigen values.

| *i* | Weighted sampling algorithms | $z=(R_0 - R_i)/SE$ | *p*-value | Holm *p*-value | Shaffer *p*-value |
|---|---|---|---|---|---|
| 1 | WRNS vs. WSPS | 4.0567 | 0.0001 | 0.0033 | 0.0033 |
| 2 | WRES vs. WSPS | 3.5496 | 0.0004 | 0.0036 | 0.0050 |
| 3 | WRNS vs. WDLAS | 3.2116 | 0.0013 | 0.0038 | 0.0050 |
| 4 | WRES vs. WDLAS | 2.7045 | 0.0068 | 0.0042 | 0.0050 |
| 5 | WRWS vs. WSPS | 2.3664 | 0.0180 | 0.0045 | 0.0050 |
| 6 | WRNS vs. WMHRW | 2.1974 | 0.0280 | 0.0050 | 0.0050 |
| 7 | WMHRW vs. WSPS | 1.8593 | 0.0630 | 0.0056 | 0.0056 |
| 8 | WRES vs. WMHRW | 1.6903 | 0.0910 | 0.0063 | 0.0063 |
| 9 | WRNS vs. WRWS | 1.6903 | 0.0910 | 0.0071 | 0.0071 |
| 10 | WRWS vs. WDLAS | 1.5213 | 0.1282 | 0.0083 | 0.0083 |
| 11 | WRES vs. WRWS | 1.1832 | 0.2367 | 0.0100 | 0.0100 |
| 12 | WMHRW vs. WDLAS | 1.0142 | 0.3105 | 0.0125 | 0.0125 |
| 13 | WDLAS vs. WSPS | 0.8452 | 0.3980 | 0.0167 | 0.0167 |
| 14 | WRWS vs. WMHRW | 0.5071 | 0.6121 | 0.0250 | 0.0250 |
| 15 | WRNS vs. WRES | 0.5071 | 0.6121 | 0.0500 | 0.0500 |

**Note:** Nemenyi's procedure rejects those hypotheses that have a *p*-value ≤ 0.0033; Holm's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0041; Shaffer's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.0033.

### 4.2.3. Experiment III

This experiment is conducted to compare the proposed algorithms with respect to their costs. To assess the cost of a sampling algorithm, a measure for the relative cost of a sampling algorithm has been defined as the cost for the total number of times that the edge weights in the graph are considered or probability of visiting each edge is calculated ($C_w$) plus the cost for total number of times that the edges in the graph are traversed or selected during the traversal of the graph ($C_t$) plus the cost for total number of operations performed during the post processing ($C_p$) divided by the total number of edges in the graph ($E$) as given by equation 13.

$$Relative\ Cost = \frac{c_1.C_w + c_2.C_t + c_3.C_p}{|E|} \tag{13}$$

where constant $c_1$ is the coefficient cost of performing weighting operation for edges, $c_2$ is the coefficient cost of traversing edges and constant $c_3$ is the coefficient cost of performing an operation during the post processing phase. In the proposed weighted sampling algorithms the cost of considering weights ($C_w$) for edges is added for each algorithm in comparison with unweighted version of sampling algorithms. For WSP and WDLA, also additional processing is added as the cost of comparisons required for ranking the traversed edges ($C_p$).

For this experiment, the sampling rate is varied from 10 to 50% with 10% interval. The comparison is performed with respect to the relative cost of the sampling algorithms defined by equation (13). The results of this experiment for different weighted sampling algorithms are presented in figure 10. As it is shown, for all sampling algorithms, increasing sampling rate results in increasing the cost. The results also indicate that for a particular given sampling rate, WRNS has the lowest cost and WDLAS has the highest cost among the proposed weighted sampling algorithms. It needs to be noted that two-phase sampling algorithms such as WSPS and WDLAS have higher cost as compared two one-phase sampling algorithms, such as WRNS, WRES, WRWS and WMHRW. Higher cost for two-phase sampling algorithms is due to the additional pre/post processing on the network, performed by the algorithm. However, two-phase sampling algorithms produce more accurate results with the expense of higher cost.

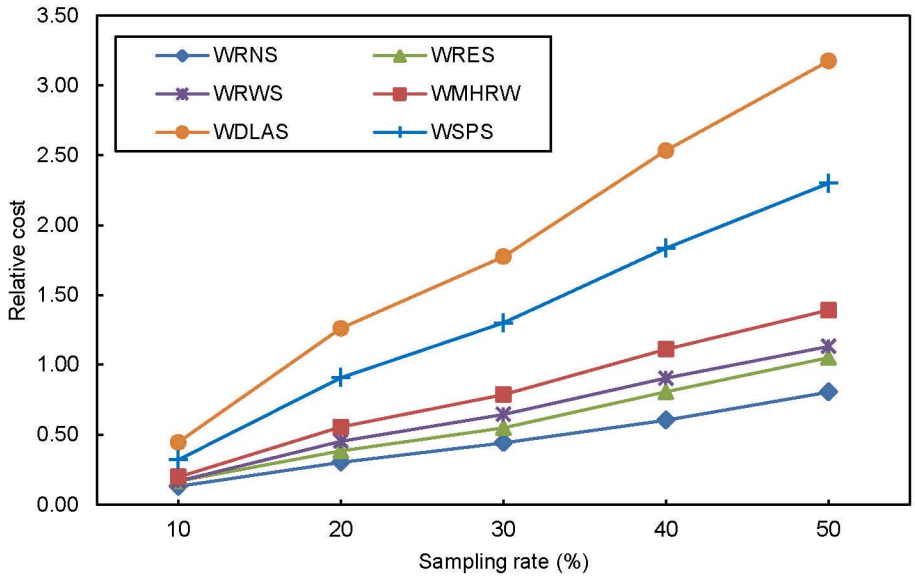
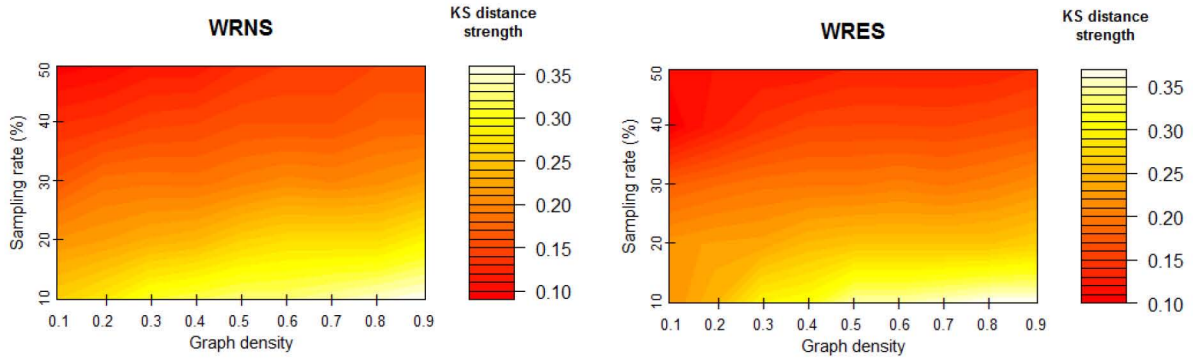


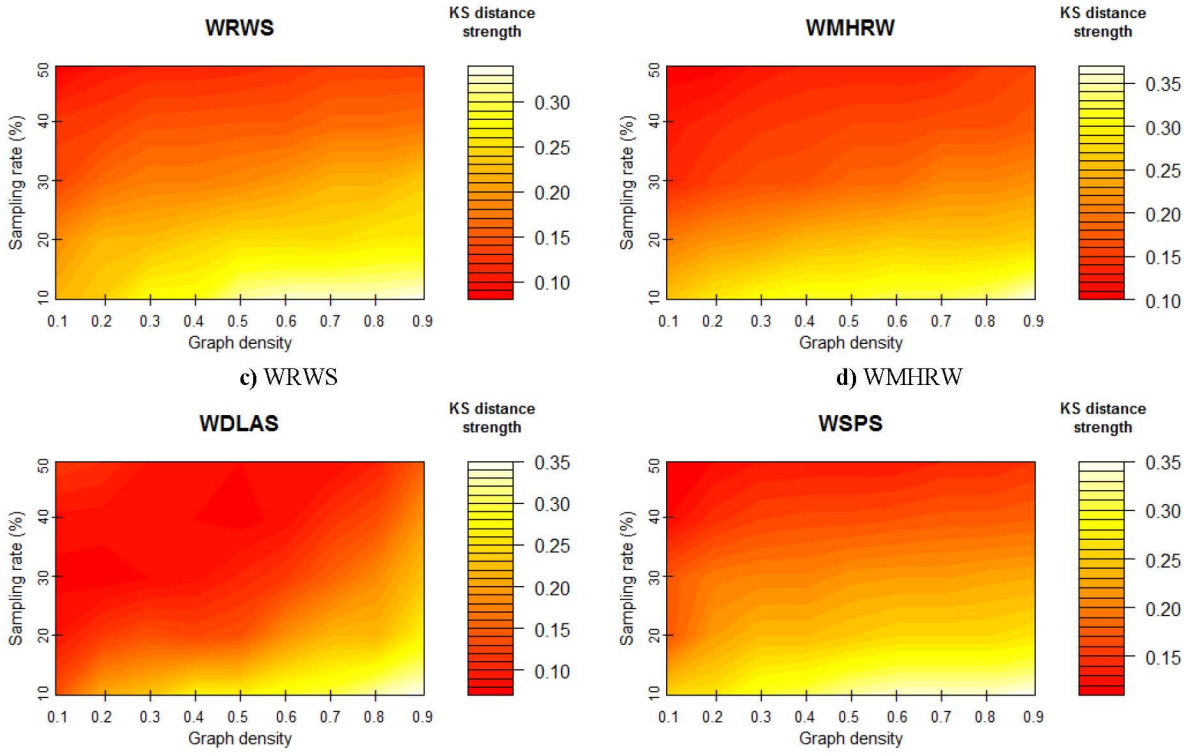

**Figure 10.** Comparing weighted sampling algorithms in terms of cost for varying sampling rate.

### 4.2.4. Experiment IV

This experiment is conducted to study the impact of graph densities on the performance of the proposed weighted sampling algorithm. For this purpose, a set of computer generated network based on the Erdős-Rényi algorithm (ER model) (Erdos and Rényi 1960) was used as synthetic network as $N \in \{5000, 10000, 20000, 50000\}$ and different densities varying from 0.2 to 0.9 with increment 0.1. To generate the weighted network instances, the random weight between 0 and 1 uniformly is selected and assigned for every edge of the synthetic ER graphs. The average results on all test networks for KS distance for strength and clustering coefficient distributions are presented in Figures 11 and 12 both as a heat map, respectively. As shown in Figures 11 and 12, for all sampling algorithms, the performance in terms of KS distance for strength and clustering coefficient distributions, improves as the sampling rate increases and the graph density decreases. The best result is obtained for high density graphs and low sampling rates and the worst results are obtained for low density graphs and high sampling rates. The results show that for high density graphs, WSPS and WDLAS outperform the other sampling algorithms in terms of KS distance for strength and clustering coefficient distributions. For low density graphs, WDLAS outperforms other sampling algorithms in terms of KS distance for strength distribution.



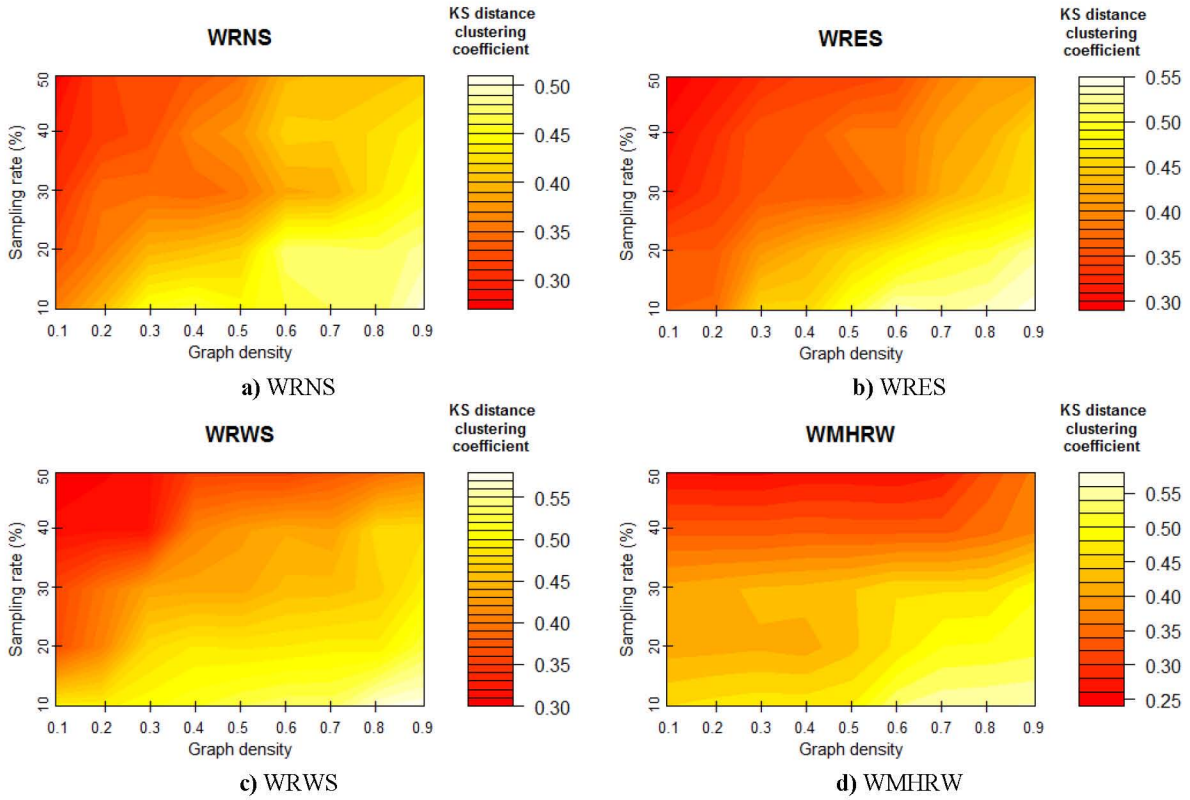

**a)** WRNS **b)** WRES




**c)** WRWS **d)** WMHRW

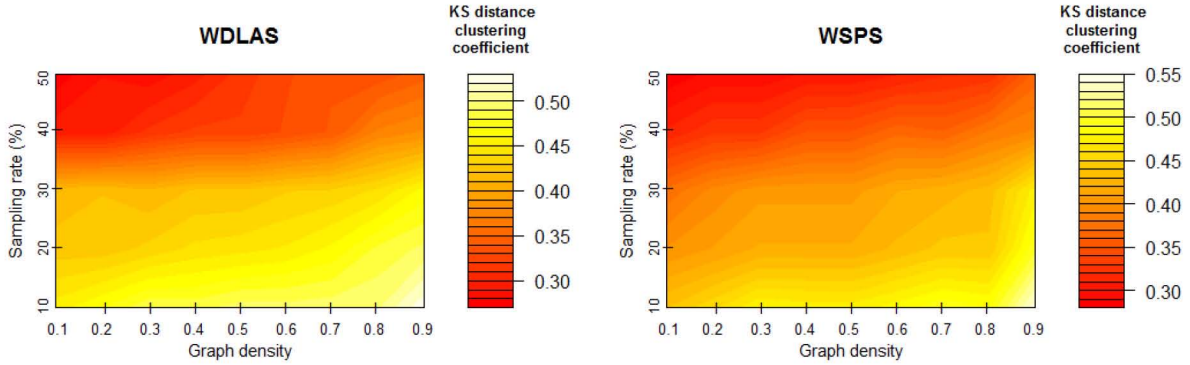
**e)** WDLAS **f)** WSPS

**Figure 11.** The relationships between graph density, sampling rate and KS distance for strength distribution for synthetic weighted graphs.




**a)** WRNS **b)** WRES

**c)** WRWS **d)** WMHRW




**e)** WDLAS **f)** WSPS

**Figure 12.** The relationships between graph density, sampling rate and KS distance for clustering coefficient distribution for synthetic weighted graphs.

# 5. Conclusion

In this paper, we introduced several network measures for weighted networks and then proposed six network sampling algorithms for weighted networks. The performance of the proposed algorithms was tested on various real weighted network benchmarks in terms of relative error, skew divergence, Pearson's correlation coefficient and the Kolmogorov-Smirnov statistic. The simulation results on real weighted networks showed that the weighted sampling algorithms significantly outperform their unweighted version of them, and also one can conclude that in general the edge weights of the weighted networks as a natural component of networks is important and the process of sampling using weighted sampling algorithms in which the edge weights of networks are considered leads to promising sampling results.

### Acknowledgements

The authors would like to thank the anonymous reviewers of this paper for their useful comments.

# 6. References

Beebe NH (2002) Nelson HF Beebe's bibliographies page. In: Nelson H. F. Beebe's Bibliographies Page. http://www.math.utah.edu/~beebe/bibliographies.html.

Blagus N, Šubelj L, Weiss G, Bajec M (2015) Sampling promotes community structure in social and information networks. Physica A: Statistical Mechanics and its Applications 432:206–215.

Chi G, Thill J-C, Tong D, et al (2016) Uncovering regional characteristics from mobile phone data: A network science approach. Papers in Regional Science doi: 10.1111/pirs.12149:1–19.

Cordeiro M, Sarmento RP, Gama J (2016) Dynamic community detection in evolving networks using locality modularity optimization. Social Network Analysis and Mining 6:15. doi: 10.1007/s13278-016-0325-1

Dall'Asta L, Barrat A, Barthélemy M, Vespignani A (2006) Vulnerability of weighted networks. Journal of Statistical Mechanics: Theory and Experiment 2006:P04006.

Dijkstra EW (1959) A note on two problems in connexion with graphs. Numerische mathematik 1:269–271.

Erdos P, Rényi A (1960) On the evolution of random graphs. Publications of the Mathematical Institute of the Hungarian Academy of Sciences 5:17–61.

Frank O (2011) Survey sampling in networks. In: The SAGE Handbook of Social Network Analysis. SAGE publications, pp 370–388

Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11:86–92.

Gao Q, Ding X, Pan F, Li W (2014) An improved sampling method of complex network. International Journal of Modern Physics C 25:1440007.

García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. Journal of Heuristics 15:617–644.

Gile KJ, Handcock MS (2010) Respondent-driven sampling: an assessment of current methodology. Sociological Methodology 40:285–327.

Gjoka M, Butts CT, Kurant M, Markopoulou A (2011) Multigraph sampling of online social networks. IEEE Journal on Selected Areas in Communications 29:1893–1905.

Gjoka M, Kurant M, Butts CT, Markopoulou A (2010) Walking in Facebook: A case study of unbiased sampling of OSNs. In: Proceedings IEEE INFOCOM 2010. San Diego, CA, pp 1–9

Guns R, Rousseau R (2014) Recommending research collaborations using link prediction and random forest classifiers. Scientometrics 101:1461–1473.

Hall BH, Jaffe AB, Trajtenberg M (2001) The NBER patent citation data file: Lessons, insights and methodological tools. National Bureau of Economic Research

Jalali ZS, Rezvanian A, Meybodi MR (2015) A two-phase sampling algorithm for social networks. In: 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI). IEEE, pp 1165–1169

Jalali ZS, Rezvanian A, Meybodi MR (2016) Social network sampling using spanning trees. International Journal of Modern Physics C 27:1650052.

Jana R, Bagchi SB (2015) Distributional Aspects of Some Statistics in Weighted Social Networks. The Journal of Mathematical Sociology 39:1–28.

Jarukasemratana S, Murata T (2015) Edge Weight Method for Community Detection on Mixed Scale-Free Networks. International Journal on Artificial Intelligence Tools 24:1–24.

Jin L, Chen Y, Hui P, et al (2011) Albatross sampling: robust and effective hybrid vertex sampling for social graphs. In: Proceedings of the 3rd ACM international workshop on MobiArch. pp 11–16

Khomami MMD, Rezvanian A, Meybodi MR (2016) Distributed learning automata-based algorithm for community detection in complex networks. International Journal of Modern Physics B 30:1650042.

Kurant M, Gjoka M, Wang Y, et al (2012) Coarse-grained topology estimation via graph sampling. In: Proceedings of the 2012 ACM workshop on Workshop on online social networks. ACM, pp 25–30

Kurant M, Markopoulou A, Thiran P (2010) On the bias of BFS (Breadth First Search). In: 2010 22nd International Teletraffic Congress (ITC). pp 1–8

Kurant M, Markopoulou A, Thiran P (2011) Towards Unbiased BFS Sampling. IEEE Journal on Selected Areas in Communications, 29:1799–1809.

Leskovec J, Faloutsos C (2006) Sampling from large graphs. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, Philadelphia, pp 631–636

Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data (TKDD) 1:1–41.

Li M, Fan Y, Wu J, Di Z (2013a) Phase transitions in Ising model induced by weight redistribution on weighted regular networks. International Journal of Modern Physics B 27:1350146.

Li P, Zhao Q, Wang H (2013b) A weighted local-world evolving network model based on the edge weights preferential selection. International Journal of Modern Physics B 27:1350039.

Li Q, Zhou T, Lü L, Chen D (2014) Identifying influential spreaders by weighted LeaderRank. Physica A: Statistical Mechanics and its Applications 404:47–55.

Li W, Cai X (2004) Statistical analysis of airport network of China. Physical Review E 69:46106.

Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology 58:1019–1031.

Lu J, Li D (2012) Sampling online social networks by random walk. In: Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research. ACM, Beijing, pp 33–40

Lü L, Zhou T (2010) Link prediction in weighted networks: The role of weak ties. EPL (Europhysics Letters) 89:18001.

Lu Z, Sun X, Wen Y, et al (2015) Algorithms and Applications for Community Detection in Weighted Networks. IEEE Transactions on Parallel and Distributed Systems 26:2916–2926.

Luo P, Li Y, Wu C, Zhang G (2015) Toward cost-efficient sampling methods. International Journal of Modern Physics C 26:1550050.

Maiya AS, Berger-Wolf TY (2010) Sampling community structure. In: Proceedings of the 19th international conference on World wide web. pp 701–710

Murai F, Ribeiro B, Towsley D, Wang P (2013) On Set Size Distribution Estimation and the Characterization of Large Networks via Sampling. IEEE Journal on Selected Areas in Communications 31:1017–1025.

Nemenyi P (1962) Distribution-free multiple comparisons. In: Biometrics. International Biometric Soc 1441 I St, Nw, Suite 700, Washington, Dc 20005-2210, p 263

Newman ME (2001) The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences 98:404–409.

Newman MEJ (2004) Analysis of weighted networks. Physical Review E 70:56131.

Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. Physical Review E 74:36104.

Opsahl T, Agneessens F, Skvoretz J (2010) Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks 32:245–251.

Opsahl T, Panzarasa P (2009) Clustering in weighted networks. Social networks 31:155–163.

Pálovics R, Benczúr AA (2015) Temporal influence over the Last.fm social network - Springer. Social Network Analysis and Mining 5:1–12.

Papagelis M, Das G, Koudas N (2013) Sampling Online Social Networks. IEEE Transactions on Knowledge and Data Engineering 25:662–676.

Park H, Moon S (2013) Sampling bias in user attribute estimation of OSNs. In: Proceedings of the 22nd international conference on World Wide Web companion. International World Wide Web Conferences Steering Committee, pp 183–184

Piña-García CA, Gu D (2013) Spiraling Facebook: an alternative Metropolis–Hastings random walk using a spiral proposal distribution. Social Network Analysis and Mining 3:1403–1415.

Rejaie R, Torkjazi M, Valafar M, Willinger W (2010) Sizing up online social networks. IEEE Network, 24:32–37.

Rezvanian A, Meybodi MR (2016a) Stochastic Graph as a Model for Social Networks. Computers in Human Behavior 64: 621–640.

Rezvanian A, Meybodi MR (2015a) Finding Maximum Clique in Stochastic Graphs Using Distributed Learning Automata. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 23:1–31.

Rezvanian A, Meybodi MR (2015b) Sampling social networks using shortest paths. Physica A: Statistical Mechanics and its Applications 424:254–268.

Rezvanian A, Meybodi MR (2016b) A new learning automata-based sampling algorithm for social networks. International Journal of Communication Systems doi: 10.1002/dac.3091:1–21.

Rezvanian A, Rahmati M, Meybodi MR (2014) Sampling from complex networks using distributed learning automata. Physica A: Statistical Mechanics and its Applications 396:224–234.

Ribeiro B, Towsley D (2010) Estimating and sampling graphs with multidimensional random walks. In: Proceedings of the 10th annual conference on Internet measurement. Melbourne, pp 390–403

Salehi M, Rabiee HR, Nabavi N, Pooya S (2011) Characterizing Twitter with Respondent-Driven Sampling. In: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC). pp 1211–1217

Salehi M, Rabiee HR, Rajabi A (2012) Sampling from complex networks with high community structures. Chaos: An Interdisciplinary Journal of Nonlinear Science 22:23126.

Saramäki J, Kivelä M, Onnela J-P, et al (2007) Generalizations of the clustering coefficient to weighted complex networks. Physical Review E 75:27105.

Saramaki J, Onnela J-P, Kertész J, Kaski K (2005) Characterizing motifs in weighted complex networks. Science of Complex Networks From Biology to the Internet and WWW 776:108–117.

Sett N, Singh SR, Nandi S (2016) Influence of edge weight on node proximity based link prediction methods: An empirical analysis. Neurocomputing 172:71–83.

Sun Y, Liu C, Zhang C-X, Zhang Z-K (2014) Epidemic spreading on weighted complex networks. Physics Letters A 378:635–640.

Tasgin M, Bingol HO (2012) Gossip on weighted networks. Advances in Complex Systems 15:1–18.

Thi DB, Ichise R, Le B (2014) Link Prediction in Social Networks Based on Local Weighted Paths. In: Future Data and Security Engineering. Springer, pp 151–163

Tong C, Lian Y, Niu J, et al (2016) A novel green algorithm for sampling complex networks. Journal of Network and Computer Applications 59:55–62.

Wang P, Zhao J, Lui J, et al (2015) Unbiased Characterization of Node Pairs over Large Graphs. ACM Transactions on Knowledge Discovery from Data (TKDD) 9:22.

Wang S-L, Tsai Y-C, Kao H-Y, et al (2013) Shortest paths anonymization on weighted graphs. International Journal of Software Engineering and Knowledge Engineering 23:65–79.

Yan X, Zhai L, Fan W (2013) C-index: A weighted network node centrality measure for collaboration competence. Journal of Informetrics 7:223–239.

Yang C-L, Kung P-H, Chen C-A, Lin S-D (2013) Semantically sampling in heterogeneous social networks. In: Proceedings of the 22nd international conference on World Wide Web companion. pp 181–182

Yarlagadda R, Pinnaka S, Etinkaya EKÇ (2015) A time-evolving weighted-graph analysis of global petroleum exchange. In: 2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM). IEEE, pp 266–273

Yoon S, Lee S, Yook SH, Kim Y (2007) Statistical properties of sampled networks by random walks. Physical Review E 75:46114.

Yoon S-H, Kim K-N, Hong J, et al (2015) A community-based sampling method using DPL for online social networks. Information Sciences 306:53–69.

Zhao SX, Rousseau R, Fred YY (2011) h-Degree as a basic measure in weighted networks. Journal of Informetrics 5:668–677.

Zheng Y, Liu F, Gong Y-W (2014) Robustness in Weighted Networks with Cluster Structure. Mathematical Problems in Engineering 2014:1–8.

Zhu M, Cao T, Jiang X (2014) Using clustering coefficient to construct weighted networks for supervised link prediction. Social Network Analysis and Mining 4:1–8.

(2016a) The University of Florida Sparse Matrix Collection. In: The University of Florida Sparse Matrix Collection. http://www.cise.ufl.edu/research/sparse/matrices.

(2016b) Pajek datasets. In: Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data.