Physica A 396 (2014) 224–234

Contents lists available at ScienceDirect

Physica A

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

Sampling from complex networks using distributed learning automata

Alireza Rezvanian [a,*], Mohammad Rahmati [b], Mohammad Reza Meybodi [a]

[a] Soft Computing Laboratory, Computer Engineering and Information Technology Department, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Hafez Ave., 424, Iran

[b] Image Processing and Pattern Recognition Laboratory, Computer Engineering and Information Technology Department, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Hafez Ave., 424, Iran

HIGHLIGHTS

- We propose a distributed learning automata based algorithm for sampling from complex networks.
- The proposed algorithm is studied on 9 popular complex networks.
- The proposed algorithm is compared with well-known sampling methods.
- The experimental results show that the proposed algorithm is a viable approach for sampling from complex networks.

ARTICLE INFO

Article history:
Received 8 July 2013
Received in revised form 6 September 2013
Available online 18 November 2013

Keywords:
Complex networks
Social networks
Network sampling
Distributed learning automata

ABSTRACT

A complex network provides a framework for modeling many real-world phenomena in the form of a network. In general, a complex network is considered as a graph of real world phenomena such as biological networks, ecological networks, technological networks, information networks and particularly social networks. Recently, major studies are reported for the characterization of social networks due to a growing trend in analysis of online social networks as dynamic complex large-scale graphs. Due to the large scale and limited access of real networks, the network model is characterized using an appropriate part of a network by sampling approaches. In this paper, a new sampling algorithm based on distributed learning automata has been proposed for sampling from complex networks. In the proposed algorithm, a set of distributed learning automata cooperate with each other in order to take appropriate samples from the given network. To investigate the performance of the proposed algorithm, several simulation experiments are conducted on well-known complex networks. Experimental results are compared with several sampling methods in terms of different measures. The experimental results demonstrate the superiority of the proposed algorithm over the others.

1. Introduction

Many real world systems such as biological, social, ecological, technological and information systems are modeled as networks, which are represented as graphs with a set of vertices (e.g. users in social networks) and edges (relationship between users in social networks). It is shown that there are universal common features in different real world networks such as small-world and scale-free properties [1,2]. In recent years, complex networks have attracted an great deal of attraction by researchers in many applications [3–5]. In Ref. [6], information in the form of periodic orbits (cycles) exists

* Corresponding author. Tel.: +98 21 6454 5120.
E-mail address: a.rezvanian@aut.ac.ir (A. Rezvanian).

0378-4371/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.physa.2013.11.015

in the network modeled as complex networks. Moreover, complex networks have provided a suitable framework for real world systems such as software systems [7]. Due to the dynamicity, complexity, and large scale of complex networks, the conventional techniques of problem solving might experience some difficulties facing real networks [8]. Therefore, a network is characterized using metrics (e.g. centrality measures) or techniques (e.g. sampling), instead of access to the whole network [9–11]. In case of online social networks, in addition to the large scale of the network, limited access due to privacy settings has arisen. So, it is impossible to fully access the whole network. In practice, researchers via sampling from networks can estimate the characteristics of given large networks [12].

A sample from graph $G = \langle V, E\rangle$, where $V(G)$ is the set of $|V| = n$ vertices and $E(G)$ denotes the edge set including $|E| = m$ edges, is a function $f : G \to S$ from the given graph G to the sample-set $S = \langle V', E'\rangle$ such that $V' \subset V$ and $E' \subset E$. It is noted that the main goal of the sampling technique is to select a smaller subgraph from G. Sampling methods [9,11,13] play a significant role in preprocessing, characterizing, and studying networks. Sampling can be used to study a small part of the networks while preserving features of the initial network. For example in online social networks, due to privacy restrictions, it could not access the whole network information, therefore, the networks are characterized by estimated small samples from a significant part of the networks. Taking into account complexity, large scale and dynamic properties of the complex networks, most traditional methods were not scalable and thus failed. Moreover, the policies of some social network websites have restricted accessibility to the whole network simultaneously [14], because of computational or privacy restrictions. Hence, there are some special complexities and challenges in sampling from complex networks which can still be discussed as new research fields [15].

In general, for studying social networks such as Facebook/Twitter, in fact there is no assumption about the probability distribution of particular measures e.g. degree distributions, whereas some researchers have to address considering special assumptions of the network *a priori* [10] for computing the biasness. In p2p networks which are highly dynamic, vertices/edges are continuously added or removed from the network, while the sampling process covers some other parts of the network [16]. Moreover, the weights of edges considered in the network are usually assumed constant, though these weights are variables over time [17]. In some cases, sampling methods not only are limited to using smaller scales in order to reduce storage space and computational complexity, but also they are used to estimate, characterize and study networks. For example, the latest estimation of indexed web pages reports 3.6 billion pages in the internet [18].

The various methods reported in the literature for sampling from graphs can be categorized into three general approaches for the sampling techniques. In the first approach, the scale of the initial graph is reduced, which is mainly used for visualizing the graph applications [19] and describing the initial graphs. In this approach, several techniques can be utilized such as clustering [20], coarse graining [21], k-core [22] and fractal based methods [23]. In the second category, the random selection is done based on either nodes or edges which often does not provide proper results since it models just a small part of the graph without considering its topological structure [15]. The third category includes crawling methods or topology based sampling methods which has attracted a great deal of attention in the literature. As the topological structure is provided relatively by the crawling methods such as Breadth-First-Search (BFS) [24], Depth-First-Search (DFS) [25], Forest Fire Sampling (FF) [26], Snowball [27], Random Walk (RW) [28], Metropolis–Hastings Random Walk (MHRW) [29], Weighted Random Walk (WRW) [30], Stratified Weighted Random Walk (SWRW) [30], and Respondent Driven Sampling (RDS) [31], they offer more effective results in comparison with random selection methods. All of these approaches are similar in their basics with their general difference being the selection strategy for a part of the graph [9].

Leskovec et al. [15] proposed a sampling method from a large graph with two goals of back in time and down scale, while several methods have been introduced and compared. The study has demonstrated that vertex selection and edge selection techniques do not provide desirable results regarding their ignorance of the correlations between selected vertices/edges. An analytical comparison between RW and BFS sampling has been presented by Kurant et al. [10] to sample from a network. Their study reveals that the degree of the graph is overstated by the BFS, while it is understated by the RW sampling. Therefore, they suggested analytical solutions to correct the bias of estimation. A practical framework for uniform sampling from users of the social network Facebook has been developed based on crawling in Ref. [16]. In this research, the advantage of unbiased estimation of MHRW and Re-WRW (RWRW) over random sampling and breadth-first-search has been addressed by comparing various approaches. RDS was analyzed in Ref. [32] to reduce the biases associated with chain referral sampling of hidden populations. And then later, sampling from Twitter using the RDS method has been reported to characterize it in Ref. [33]. They have shown, through experimental examinations on Twitter, the lower error in RDS than that of MHRW. Analysis of the random walk method has been developed by Cooper et al. [34], where the authors have tried to sample from the high degree vertices and similar graphs regarding the power law distribution. Cumulative distribution of degrees is estimated via sampling based on tracerouting and some methods were studied for eliminating bias of the high degrees [35]. Ribeiro et al. proposed a sampling method, called Frontier sampling. It is developed from the traditional random walk which used several dependent random walks. The frontier sampling outperforms the conventional random walk and generates small errors in sparse graphs. According to the diversity communication between users of the social networks, a multi-graph has been introduced using random walk [11] and the results of its simulation indicate improvement of the proposed method by Gjoka. Random jump in MHRW for unbiased estimation has been proposed and it also prevents being trapped in local structures in Ref. [36] by Jin et al.

Variety in social network models makes various kinds of modeling possible. A modular structure has been studied in complex networks with sampling from the network being implemented based on identifying the communities by Maiya et al. [37]. In other research, structure of a bipartite graph has been considered in Ref. [38] for some social networks, then

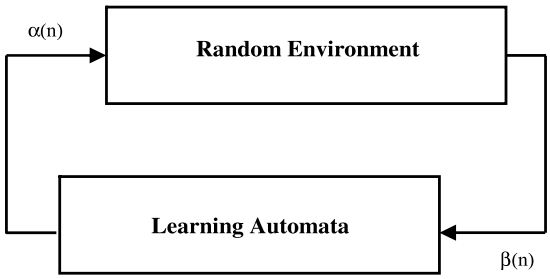

Fig. 1. The relationship between the learning automaton and its random environment.

MHRW has been utilized to suggest sampling in this graph. Sampling from directed graphs has been proposed in Ref. [39] using random walk. A directed heterogeneous graph [40] is suggested by Yang et al. in order to sample semantically. They show that their sampling technique preserves the relational profile property. Sampling by crawling the edges has been discussed in Ref. [41] by the idea of page rank which provides more significant results in comparison with RDS.

In this paper, an algorithm based on distributed learning automata is proposed for sampling from complex networks. In the proposed algorithm, a learning automaton is assigned to each vertex of the graph, in which a set of distributed learning automata cooperate with each other to take promising samples from the important part of the graph. In the proposed algorithm, if the selected vertex is evaluated well, the selected vertex is rewarded, and penalized otherwise. The proposed algorithm iteratively repeats until it proceeds to predefined criteria. The proposed algorithm is compared with several sampling methods and the obtained results show the superiority of the proposed algorithm over the others in terms of several criteria.

The rest of this paper is organized as follows. Section 2 introduces the learning automata and distributed learning automata. In Section 3, the proposed algorithm for sampling from complex networks is described. The performance of the proposed algorithm is investigated through the simulation experiments on the standard graph of complex networks in Section 4, and finally Section 5 concludes this paper.

2. Learning automata

A learning automaton [42,43] is an adaptive decision-making unit that improves its performance by learning how to choose the optimal action from a finite set of allowed actions through repeated interactions with a random environment. The action is chosen at random based on a probability distribution kept over the action-set and at each instant the given action serves as the input to the random environment. The environment responds to the taken action in turn with a reinforcement signal. The action probability vector is updated based on the reinforcement feedback from the environment. The objective of a learning automaton is to find the optimal action from the action-set so that the average penalty received from the environment is minimized.

The environment can be described by a triple $E \equiv \{\alpha, \beta, c\}$, where $\alpha \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ represents the finite set of the inputs, $\beta \equiv \{\beta_1, \beta_2, \ldots, \beta_m\}$ denotes the set of the values that can be taken by the reinforcement signal, and $c \equiv \{c_1, c_2, \ldots, c_m\}$ denotes the set of the penalty probabilities, where the element c_i is associated with the given action α_i. If the penalty probabilities are constant, the random environment is said to be a stationary random environment, and if they vary with time, the environment is called a non-stationary environment. The environments depending on the nature of the reinforcement signal β can be classified into P-*model*, Q-*model* and S-*model*. The environments in which the reinforcement signal can only take two binary values 0 and 1 are referred to as P-model environments. Another class of the environment allows a finite number of values in the interval [0, 1] can be taken by the reinforcement signal. Such an environment is referred to as a Q-model environment. In S-model environments, the reinforcement signal lies in the interval $[a, b]$. The relationship between the learning automaton and its random environment has been shown in Fig. 1 [42].

Learning automata can be classified into two main families [42]: fixed structure learning automata and variable structure learning automata. Variable structure learning automata are represented by a triple $\langle \beta, \alpha, T \rangle$, where β is the set of inputs, α is the set of actions, and T is the learning algorithm. The learning algorithm is a recurrence relation which is used to modify the action probability vector. Let $\alpha(k)$ and $p(k)$ denote the action chosen at instant k and the action probability vector on which the chosen action is based, respectively. The recurrence equation shown by (1) and (2) is a linear learning algorithm by which the action probability vector p is updated. Let $\alpha(k)$ be the action chosen by the automaton at instant k

$$p_j(k+1) = \begin{cases} p_j(k) + a[1 - p_j(k)] & j = i \\ (1-a)p_j(k) & \forall j \neq i \end{cases} \tag{1}$$

when the taken action is rewarded by the environment (i.e. $\beta(n) = 0$) and

$$p_j(k+1) = \begin{cases} (1-b)p_j(k) & j = i \\ \left(\dfrac{b}{r-1}\right) + (1-b)p_j(k) & \forall j \neq i \end{cases} \tag{2}$$

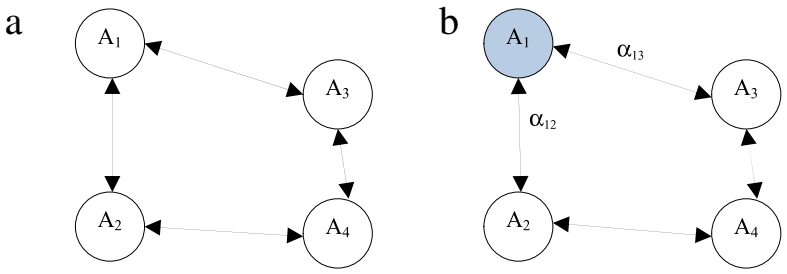

Fig. 2. Distributed learning automata.

when the taken action is penalized by the environment (i.e. $\beta(n) = 1$). r is the number of actions that can be chosen by the automaton, $a(k)$ and $b(k)$ denote the reward and penalty parameters and determine the amount of increases and decreases of the action probabilities, respectively. If $a(k) = b(k)$, the recurrence equations (1) and (2) are called the linear reward–penalty (L_{R-P}) algorithm, if $a(k) \gg b(k)$ the given equations are called the linear reward-ε penalty ($L_{R-\varepsilon P}$), and finally if $b(k) = 0$ they are called the linear reward–Inaction (L_{R-I}). In the latter case, the action probability vectors remain unchanged when the taken action is penalized by the environment.

Learning automata have been found to be useful in systems where incomplete information about the environment, wherein the system operates, exists. Learning automata are also proved to perform well in dynamic environments. It has been shown in Ref. [44] that the learning automata are capable of solving the distributed problems. Recently, several learning automata based approaches have been presented for improving the performance of many applications [45–49].

2.1. Distributed learning automata

A Distributed learning automata (DLA) [50–53] shown in Fig. 2 is a network of interconnected learning automata which collectively cooperate to solve a particular problem. Formally, a DLA can be defined by a quadruple $\langle A, E, T, A_0\rangle$, where $A = \{A_1, A_2, \ldots, A_n\}$ is the set of learning automata, $E \subset A \times A$ is the set of the edges in which edge $e_{(i,j)}$ corresponds to the action α_{ij} of the automaton A_i, T is the set of learning schemes with which the learning automata update their action probability vectors, and A_0 is the root automaton of the DLA from which the automaton activation is started.

The operation of a DLA can be described as follows. At first, the root automaton randomly chooses one of its outgoing edges (actions) according to its action probabilities and activates the learning automaton at the other end of the selected edge. The activated automaton also randomly selects an action which results in activation of another automaton. The process of choosing the actions and activating the automata is continued until a leaf automaton (an automaton which interacts with the environment) is reached. The chosen actions, along the path induced by the activated automata between the root and leaf, are applied to the random environment. The environment evaluates the applied actions and emits a reinforcement signal to the DLA. The activated learning automata along the chosen path update their action probability vectors on the basis of the reinforcement signal by using the learning schemes. The paths from the unique root automaton to one of the leaf automata are selected until the probability with which one of the chosen paths is close enough to unity. Each DLA has exactly one root automaton which is always activated, and at least one leaf automaton which is activated probabilistically. For example in Fig. 2, every automaton has two actions. If automaton A_1 selects α_3 from its action set, then it will be the activated automaton of A_3. Afterwards, the automaton of A_3 will choose one of its possible actions and so on.

3. Proposed sampling algorithm based on distributed learning automata

In the proposed algorithm, first, each vertex of the graph is equipped with a learning automaton, and as a result, a network of learning automata isomorphic to the graph is initially constructed. The set of distributed learning automata is defined by a tuple $\langle A, \alpha\rangle$, where $A \equiv \{A_1, A_2, \ldots, A_n\}$ is the set of learning automata corresponding to the set of vertices, and $\alpha \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ denotes the set of actions in which $\alpha_i = \{\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^{r_i}\}$ described the set of actions that can be taken by learning automaton A_i. In the proposed algorithm, the action set of each learning automaton (e.g. A_i) is choosing one of the neighbors which are initialized equally as

$$p_{ij} = \frac{1}{d(v_i)} \tag{3}$$

where $d(v_i)$ is the degree of vertex v_i.

At the beginning of the algorithm, all learning automata are considered inactive. Randomly, a vertex is selected and activated (e.g. A_i), the activated automaton selected one of its neighbors (A_j) according to the probability vector of A_i. The

Algorithm 1: proposed algorithm for sampling from complex networks using distributed learning automata

Input: Graph $G=\langle V, E\rangle$, Sample size k, convergence threshold τ, maximum iteration T
Output: The set of sample vertices
Assumptions
Assign an automaton A_i to each vertex v_i
Let $\alpha_i=\{\alpha_i^1, \alpha_i^2, \ldots \alpha_i^{ri}\}$ denotes the action set of automaton A_i
Let N denotes the number of vertices and M denotes the number of edges
Let δ denotes the list of sampled vertices and initially set to $\{\}$
Select node s as seed node randomly
Begin
Disables all learning automata
While stopping criteria are not meet (τ or T) **Do**
Enable A_s
Choose an action by A_s as i according to eq.(4)
If (A_s visited before)
The chosen action by the learning automaton is rewarded
End If
$s \leftarrow v_i$
disable A_i
End While
Sort vertices
$\delta \leftarrow$k-top(δ)
End Algorithm

Fig. 3. The pseudo code of the proposed sampling algorithm based on distributed learning automata.

degree of vertices has also been considered in the process of selecting vertices as

$$p^j(t+1) = \frac{\left(p_i^j(t) \cdot d(v_i)^{-1}\right)}{\sum_{i=1}^{r}\left(p_i^j(t) \cdot d(v_i)^{-1}\right)} \tag{4}$$

where $p_i^j(t)$ represents the probability of selected action α_i by learning automaton A_j in time t, and $d(v_i)$ denotes the degree of vertex v_i. In fact, the probability of actions with greater probability and smaller degree is increased.

Having selected the new vertex, the previous vertex is deactivated while the new vertex is activated. In the proposed algorithm, if the selected vertex has been visited before, the corresponding automaton is rewarded, otherwise penalized. This process is iteratively repeated until a predefined number of iterations or until it exceeds a predefined threshold of convergence. Threshold of convergence is defined by the product of the action probability vector of the learning automaton as follows

$$\tau = \prod_{j \in S} \arg\max\{p^j(t)\} \tag{5}$$

where t denotes the iteration number of the algorithm and S is the vertex set of samples. p^j represents the probability vector of automaton A_j.
At the end of this process, k-top vertices which belong to the highest probability; will be taken as desirable samples.

The pseudo code of the proposed sampling algorithm based on the distributed learning automaton is shown in Fig. 3.

4. Simulation results

To investigate the performance of the proposed algorithm, several simulation experiments are employed on the well-known data sets of complex networks. In the simulation experiments, the proposed algorithm is compared with several sampling methods from complex networks. The description of networks has been listed in Table 1. In all the experiments

Table 1
Description of standard data set of the complex networks.

Network	Node	Link	Type
Karate	34	78	Social
Football	115	613	Social
Dolphin	62	159	Social
Adjnoun	112	425	Word
C. elegans	453	2025	Biological
Airport	332	2126	Technological
Email	1133	5451	Social
Glossary	72	122	Words
Jazz	198	2742	Social

presented in the paper, the learning scheme is L_{R-I} and the learning rate is set to 0.01. The threshold convergence τ is set to 0.9 and the predefined iteration for the stopping algorithm is set to 1000.

There is also the visualization of given complex networks in the simulation which are depicted in Fig. 4.

4.1. Evaluation criteria

In this paper, the results of the proposed algorithm in comparison with alternative methods are reported in terms of *Kolmogorov–Smirnov Test* (K–S test), *Relative Error* (RE) and *Normalized Root Mean Square Error* (NRMSE) according to the real and estimated values of the clustering coefficient and degree distribution of vertices. These evaluation criteria are described as follows.

4.1.1. Kolmogorov–Smirnov test (K–S test)

The Kolmogorov–Smirnov D-Statistic is one of the statistical test methods used for assessing the distance between two cumulative distribution functions (CDF). D is a measure for acceptability between the original distribution and estimated distribution. The result of this test can be relatively employed for comparison. In other words, the result of this test is a value between 0 and 1. The values close to 0 indicate both distributions will have a greater similarity; and the values close to 1 indicate the two distributions will show a greater discrepancy [54]. This measure has been defined as

$$D = \max |F'(x) - F(x)| \tag{6}$$

where F and F' denote the Cumulative Distribution Function (CDF) of the 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.

4.1.2. Relative Error (RE)

Relative Error (RE) can be employed to assess the accuracy of the results, which is calculated by this equation below:

$$\mathrm{RE} = \frac{\left|\theta - \hat{\theta}\right|}{\theta} \tag{7}$$

where θ and $\hat{\theta}$ denote the values of the real and estimated parameters (e.g. real clustering coefficient and estimated clustering coefficient) in the obtained samples [9].

4.1.3. Normalized Root Mean Square Error (NMSE)

Another measure used in this regard is the Normalized Root Mean Square Error (NMSE) for the clustering coefficient which is given by the following equation [55,12]:

$$\mathrm{NMSE} = \frac{\sqrt{\left(E|C - \hat{C}|\right)^2}}{C} \tag{8}$$

where C and $\hat{C}$ are the values of the real and estimated clustering coefficients in the extracted samples.

4.2. Experimental results

In this paper, all experiments were launched on a system with a hardware configuration of Intel® Core i5 U520 1.07 GHz processor and 2 GB RAM. The linear learning algorithm L_{R-I} was used for the learning automaton and the learning parameter

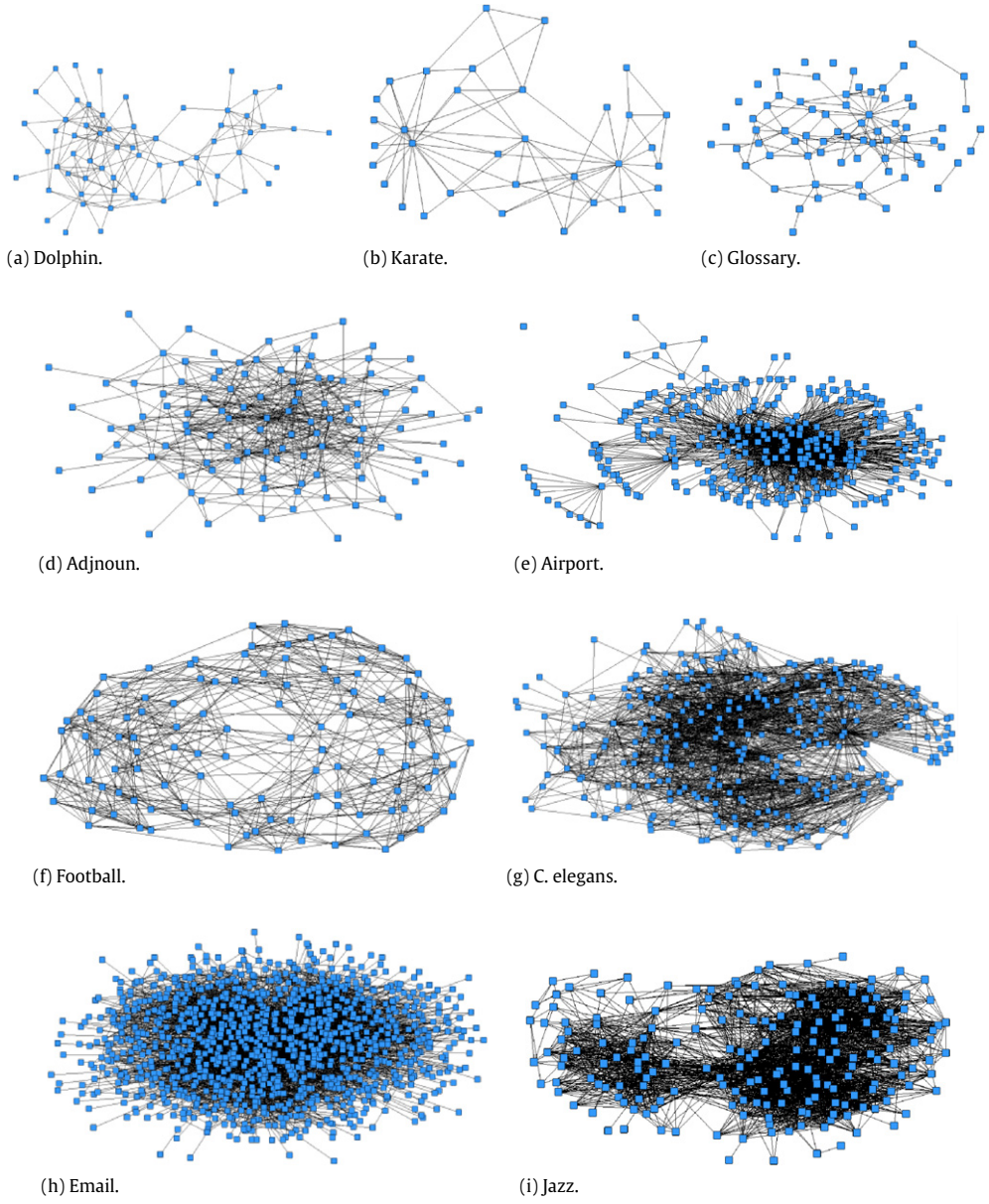

Fig. 4. Visualization of standard data set of the complex networks.

was taken as 0.01 for the first experiment. The proposed sampling algorithm was evaluated through sampling by learning automata as SLA in comparison with random vertex sampling as RVS, Random Edge Sampling as RES, Random Walk Sampling as RWS, and Respondent Driven Sampling as RDS for the sampling rate of 0.15%. In the first experiment, the performance of the proposed algorithm and other popular algorithms is investigated. The average results of varying sampling rate (0.15% and 25%) are summarized in Tables 2 and 3 for 30 independent runs in terms of the K–S test for degree distribution $\langle k \rangle$.

By focusing on the result of Tables 2 and 3, it can be inferred from most networks that those with increasing sampling rates have a noticeable impact on the K–S test. Moreover, the results have shown the superiority of the proposed algorithm except RDS for Football, C. elegans and Jazz for lower sampling rate and C. elegans for higher sampling rate.

In the second experiment, the impact of varying sampling rate is studied and compared with other algorithms in terms of relative error (RE) for clustering coefficient (CC). Fig. 5 presents RE values for CC as a bar chart diagram for varying sampling rates (15%–35%).

As shown in Fig. 5, the relative error decreases by increasing the sampling rate in most cases. Besides, for the networks with smaller size (e.g. Karate), the relative error is less than those networks with larger size (e.g. C. elegans).

Table 2
Results of *K–S test* for degree distribution $\langle k \rangle$ in the sampling methods for sampling rate = 0.15%.

Networks	Methods				
	RVS	RES	RWS	RDS	SLA
Karate	0.64	0.40	0.42	0.35	**0.34**
Football	0.57	0.49	0.38	0.19	0.19
Dolphin	0.58	0.69	0.66	0.44	**0.41**
C. elegans	0.18	0.19	0.17	**0.06**	0.07
Airport	0.36	0.35	0.34	0.20	**0.19**
Jazz	0.60	0.68	0.57	**0.35**	0.37
Adjnoun	0.36	0.35	0.34	0.23	**0.19**
Glossary	0.31	0.38	0.31	0.25	**0.22**
Email	0.39	0.36	0.34	0.29	**0.28**

Table 3
Results of *K–S test* for degree distribution $\langle k \rangle$ in the sampling methods for sampling rate = 0.25%.

Networks	Methods				
	RVS	RES	RWS	RDS	SLA
Karate	0.48	0.36	0.36	0.31	**0.29**
Football	0.51	0.42	0.34	0.18	**0.17**
Dolphin	0.52	0.61	0.59	0.39	**0.37**
C. elegans	0.15	0.16	0.14	0.06	0.06
Airport	0.33	0.34	0.31	0.18	**0.17**
Jazz	0.54	0.61	0.48	0.34	**0.33**
Adjnoun	0.36	0.35	0.34	0.21	**0.19**
Glossary	0.26	0.31	0.29	0.22	**0.20**
Email	0.34	0.32	0.29	0.27	**0.26**

Table 4
Results of *K–S test* for degree distribution $\langle k \rangle$ in the proposed sampling algorithm for sampling rate = 0.15%.

Learning parameter	Networks				
	Karate	Football	Dolphin	Adjnoun	Glossary
0.01	0.3416	**0.2478**	0.4753	0.2787	**0.2429**
0.02	0.3891	0.2942	0.4524	0.2650	0.2954
0.03	0.3443	0.2855	0.4451	0.2595	0.2870
0.04	**0.3208**	0.2826	0.4643	0.2681	0.2801
0.05	0.3344	0.2681	**0.4297**	0.2802	0.2862
0.06	0.3680	0.2971	0.4961	0.2660	0.2798
0.07	0.3332	0.2884	0.4786	0.2594	0.2790
0.08	0.4040	0.2681	0.4922	0.2572	0.2388
0.09	0.3542	0.2594	0.4765	**0.2439**	0.2732
0.10	0.3817	0.2884	0.4663	0.2479	0.2875

In the third experiment, the effect of varying sampling rate on the proposed algorithm for NMSE of the clustering coefficient has been depicted in Fig. 6 for different networks.

In general, according to Fig. 6, NMSE has a descending trend with increasing the number of samples. In the case of small networks, a greater value of NMSE is noticed for a few samples in comparison with a large number of them.

In the last experiment, the impact of learning rate for learning automata in the proposed algorithm is investigated. It is important that in the learning automata based algorithm, the convergence of the algorithm depends on an appropriate value of the learning parameter. Thus, the effect of different learning parameters on the proposed algorithm for 30 independent runs is listed in Table 4 for selected complex networks in terms of the K–S test for degree distribution $\langle k \rangle$.

It can be inferred from Table 4 that certain values (e.g. low value) of the learning parameter do not show a considerable impact on the accuracy of results, while much better results are different for each network. Therefore, the proposed method just requires a parameter tuning for the learning parameter. The proposed algorithm is simple and easy to implement with appropriate results in most cases. As previously demonstrated in the simulation results, the proposed algorithm has outperformed other approaches. Nevertheless, some further improvements can still be considered in this algorithm for more future research such as selecting a targeted initial node (nodes with higher values of centrality), selecting a node after several runs of crawling, selecting several types of neighborhood nodes, and postponing the learning process.

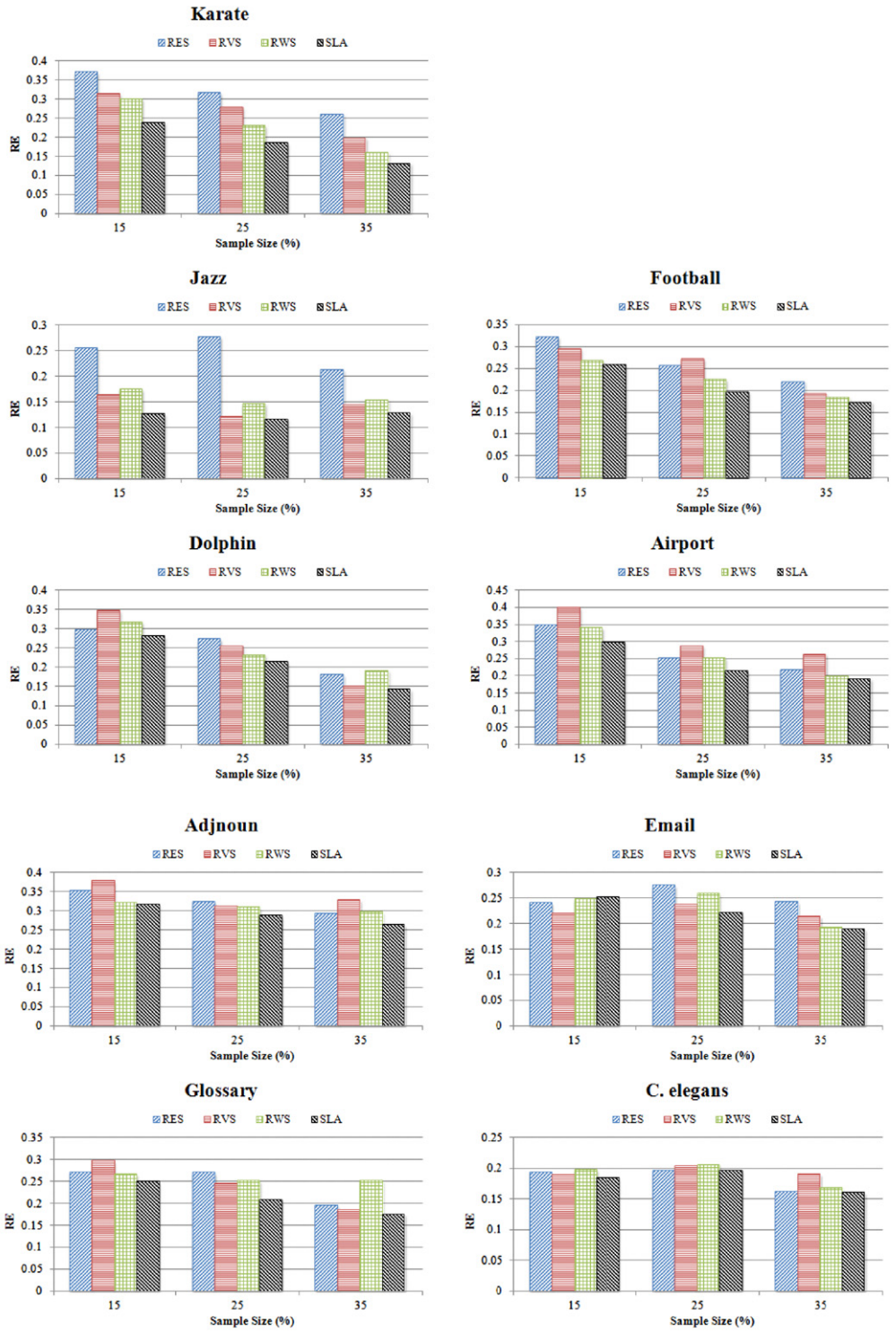

Fig. 5. Relative error of *CC* for varying sampling rates (15%, 25% and 35% respectively from left).

5. Conclusion

Sampling methods are used for studying, characterizing, describing and visualizing the dynamical complex networks. In this paper, an algorithm based on a distributed learning automaton was proposed for sampling from a complex network. In

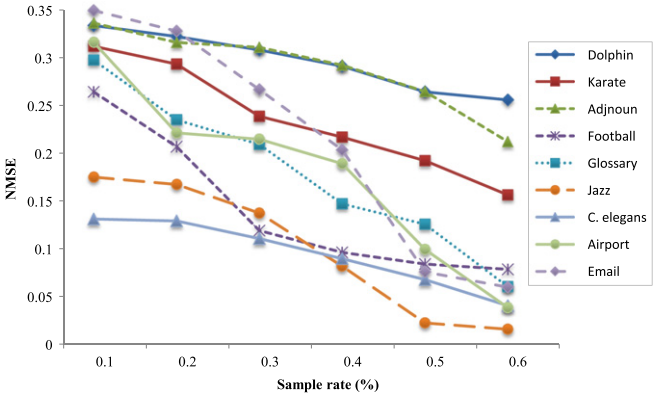

Fig. 6. Varying sampling rate for NMSE results of CC on the proposed algorithm for different networks.

the proposed algorithm, a set of learning automata were assigned onto vertices of the given graph and then the promising part of the graph was sampled based on the cooperation between distributed learning automata. The performance of the proposed sampling algorithm was empirically investigated against the well-known sampling techniques on various complex network benchmarks. According to the obtained results in several simulation experiments, the proposed algorithm outperforms the other algorithms in terms of the K–S test, RE and NMSE in most cases.

References

[1] D.J. Watts, S.H. Strogatz, Collective dynamics of "small-world" networks, Nature 393 (1998) 440–442.
[2] A.L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509–512.
[3] S. Hu, H. Yang, B. Cai, C. Yang, Research on spatial economic structure for different economic sectors from a perspective of a complex network, Physica A 392 (2013) 3682–3697.
[4] J. Tang, Y. Wang, F. Liu, Characterizing traffic time series based on complex network theory, Physica A 392 (2013) 4192–4201.
[5] G. Zhang, Z. Li, B. Zhang, W.A. Halang, Understanding the cascading failures in Indian power grids with complex networks theory, Physica A 392 (2013) 3273–3280.
[6] I. Sorkhoh, K.A. Mahdi, M. Safar, Estimation algorithm for counting periodic orbits in complex social networks, Inf. Syst. Front. 15 (2013) 193–202.
[7] J. Ma, D. Zeng, H. Zhao, Modeling the growth of complex software function dependency networks, Inf. Syst. Front. 14 (2012) 301–315.
[8] L.F. Costa, F.A. Rodrigues, G. Travieso, P.R.V. Boas, Characterization of complex networks: a survey of measurements, Adv. Phys. 56 (2007) 167–242.
[9] M. Papagelis, G. Das, N. Koudas, Sampling Online social networks, IEEE Trans. Knowl. Data Eng. 25 (2013) 662–676.
[10] M. Kurant, A. Markopoulou, P. Thiran, Towards unbiased BFS sampling, IEEE J. Sel. Areas Commun. 29 (2011) 1799–1809.
[11] M. Gjoka, C.T. Butts, M. Kurant, A. Markopoulou, Multigraph sampling of online social networks, IEEE J. Sel. Areas Commun. 29 (2011) 1893–1905.
[12] F. Murai, B. Ribeiro, D. Towsley, P. Wang, On set size distribution estimation and the characterization of large networks via sampling, IEEE J. Sel. Areas Commun. 31 (2013) 1017–1025.
[13] E. Volz, D.D. Heckathorn, Probability based estimation theory for respondent driven sampling, J. Off. Stat.-Stockholm 24 (2008) 79.
[14] M. Huisman, Imputation of missing network data: some simple procedures, J. Soc. Struct. 10 (2009) 1–29.
[15] J. Leskovec, C. Faloutsos, Sampling from large graphs, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 631–636.
[16] M. Gjoka, M. Kurant, C.T. Butts, A. Markopoulou, Walking in facebook: a case study of unbiased sampling of OSNs, in: 2010 Proceedings IEEE INFOCOM, 2010, pp. 1–9.
[17] E. Gilbert, K. Karahalios, Predicting tie strength with social media, in: Proceedings of the 27th International Conference on Human Factors in Computing Systems, 2009, pp. 211–220.
[18] The size of the World Wide Web (The Internet), 2013. Online: http://www.worldwidewebsize.com/.
[19] Y. Jia, J. Hoberock, M. Garland, J. Hart, On the visualization of social and other scale-free networks, IEEE Trans. Vis. Comput. Graphics 14 (2008) 1285–1292.
[20] S. Lafon, A.B. Lee, Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1393–1403.
[21] A. Zeng, L. Lü, Coarse graining for synchronization in directed networks, Phys. Rev. E 83 (2011) 056123.
[22] S.N. Dorogovtsev, A. Goltsev, J.F.F. Mendes, *K*-core organization of complex networks, Phys. Rev. Lett. 96 (2006) 40601.
[23] J.S. Kim, K.I. Goh, B. Kahng, D. Kim, Fractality and self-similarity in scale-free networks, New J. Phys. 9 (2007) 177.
[24] S.H. Lee, P.J. Kim, H. Jeong, Statistical properties of sampled networks, Phys. Rev. E 73 (2006) 016102.
[25] S. Even, Graph Algorithms, second ed., Cambridge University Press, 2011.
[26] M. Kurant, A. Markopoulou, P. Thiran, On the bias of BFS (Breadth First Search), in: 2010 22nd International Teletraffic Congress, ITC, 2010, pp. 1–8.
[27] O. Frank, Survey sampling in networks, in: The SAGE Handbook of Social Network Analysis, SAGE publications, 2011, pp. 370–388.
[28] S. Yoon, S. Lee, S.H. Yook, Y. Kim, Statistical properties of sampled networks by random walks, Phys. Rev. E 75 (2007) 046114.
[29] C.H. Lee, X. Xu, D.Y. Eun, Beyond random walk and Metropolis–Hastings samplers: why you should not backtrack for unbiased graph sampling, in: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, 2012, pp. 319–330.
[30] M. Kurant, M. Gjoka, C.T. Butts, A. Markopoulou, Walking on a graph with a magnifying glass, in: Proceedings of ACM SIGMETRICS, 2011, pp. 1–12.
[31] S. Goel, M.J. Salganik, Assessing respondent-driven sampling, Proc. Natl. Acad. Sci. 107 (2010) 6743–6747.
[32] K.J. Gile, M.S. Handcock, Respondent-driven sampling: an assessment of current methodology, Sociol. Methodol. 40 (2010) 285–327.
[33] M. Salehi, H.R. Rabiee, N. Nabavi, S. Pooya, Characterizing twitter with respondent-driven sampling, in: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, DASC, 2011, pp. 1211–1217.
[34] C. Cooper, T. Radzik, Y. Siantos, A fast algorithm to find all high degree vertices in power law graphs, in: Proceedings of the 21st International Conference Companion on World Wide Web, 2012, pp. 1007–1016.
[35] B. Ribeiro, D. Towsley, Estimating and sampling graphs with multidimensional random walks, in: Proceedings of the 10th Annual Conference on Internet Measurement, 2010, pp. 390–403.

[36] L. Jin, Y. Chen, P. Hui, C. Ding, T. Wang, A.V. Vasilakos, et al. Albatross sampling: robust and effective hybrid vertex sampling for social graphs, in: Proceedings of the 3rd ACM International Workshop on MobiArch, 2011, pp. 11–16.
[37] A.S. Maiya, T.Y. Berger-Wolf, Sampling community structure, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 701–710.
[38] J. Wang, Y. Guo, Unbiased sampling of bipartite graph, in: 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC, 2011, pp. 357–360.
[39] B. Ribeiro, P. Wang, F. Murai, D. Towsley, Sampling directed graphs with random walks, in: Proceedings IEEE INFOCOM, 2012, pp. 1692–1700.
[40] C.-L. Yang, P.-H. Kung, C.-A. Chen, S.-D. Lin, Semantically sampling in heterogeneous social networks, in: Proceedings of the 22nd International Conference on World Wide Web Companion, 2013, pp. 181–182.
[41] M. Salehi, H.R. Rabiee, A. Rajabi, Sampling from complex networks with high community structures, Chaos 22 (2012) 023126.
[42] K.S. Narendra, M.A.L. Thathachar, Learning Automata: An Introduction, Prentice-Hall, 1989.
[43] M.A.L. Thathachar, P.S. Sastry, Networks of Learning Automata: Techniques for Online Stochastic Optimization, Kluwer Academic Publishers, 2004.
[44] J. Akbari Torkestani, M.R. Meybodi, Finding minimum weight connected dominating set in stochastic graph based on learning automata, Inform. Sci. 200 (2012) 57–77.
[45] A. Rezvanian, M.R. Meybodi, An adaptive mutation operator for artificial immune network using learning automata in dynamic environments, in: Proceedings of the 2010 Second World Congress on Nature and Biologically Inspired Computing, NaBIC, 2010: pp. 479–483.
[46] A. Rezvanian, M.R. Meybodi, LACAIS: learning automata based cooperative artificial immune system for function optimization, in: Contemporary Computing, Springer, Berlin, Heidelberg, 2010, pp. 64–75.
[47] A. Rezvanian, M.R. Meybodi, T. Kim, Tracking extrema in dynamic environments using a learning automata-based immune algorithm, in: Grid and Distributed Computing, Control and Automation, Springer, Berlin, Heidelberg, 2010, pp. 216–225.
[48] J. Akbari Torkestani, A highly reliable and parallelizable data distribution scheme for data grids, Future Gener. Comput. Syst. 29 (2013) 509–519.
[49] F. Amiri, N. Yazdani, H. Faili, A. Rezvanian, A novel community detection algorithm for privacy preservation in social networks, in: A. Abraham (Ed.), Intelligent Informatics, 2013, pp. 443–450.
[50] H. Beigy, M.R. Meybodi, Utilizing distributed learning automata to solve stochastic shortest path problems, Int. J. Uncertain. Fuzz. 14 (2006) 591.
[51] J. Akbari Torkestani, M.R. Meybodi, Clustering the wireless Ad Hoc networks: a distributed learning automata approach, J. Parallel Distrib. Comput. 70 (2010) 394–405.
[52] R. Forsati, M.R. Meybodi, Effective page recommendation algorithms based on distributed learning automata and weighted association rules, Expert Syst. Appl. 37 (2010) 1316–1330.
[53] M. Soleimani-Pouri, A. Rezvanian, M.R. Meybodi, Solving maximum clique problem in stochastic graphs using learning automata, in: 2012 Fourth International Conference on Computational Aspects of Social Networks, CASoN, 2012, pp. 115–119.
[54] M.L. Goldstein, S.A. Morris, G.G. Yen, Problems with fitting to the power-law distribution, Eur. Phys. J. B 41 (2004) 255–258.
[55] Z. Bar-Yossef, M. Gurevich, Random sampling from a search engine's index, J. ACM 55 (2008) 24:1–24:74.