4th International Conference on Knowledge-Based Engineering and Innovation (KBEI-2017)
Dec. 22th, 2017
(Iran University of Science and Technology) – Tehran, Iran

An Algorithm for Weighted Positive Influence Dominating Set Based on Learning Automata

Mohammad Mehdi Daliri Khomami
Soft computing laboratory, Department of Computer Engineering and Information Technology
Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
m.daliri@aut.ac.ir

Maryam Amir Haeri
Department of Computer Engineering & IT, Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
haeri@aut.ac.ir

Mohammad Reza Meybodi
Soft computing laboratory, Department of Computer Engineering and Information Technology, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
mmeybodi@aut.ac.ir

Ali Mohammad Saghiri
Soft computing laboratory, Department of Computer Engineering and Information Technology, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
a_m_saghiri@aut.ac.ir

Abstract— The problem of Influence Maximization (IM) in social network refers to determining a set of nodes that can maximize the spread of influence. IM problem has been applied in many domains such as marketing, advertising and public opinion monitoring. In recent years, different type of algorithms reported in the literature. A type of algorithms for this problem is based on dominating set. In these algorithms, the IM problem is considered as a version of dominating set problem. A limitation of most of these algorithms, the graph of the social network is assumed to be a un-weighted graph which is not a realistic assumption. The other drawback of these algorithms is that they generally omit the crucial characteristics of the social networks such as different level of influence and dynamic interaction among individuals. In order to solve these problems and based on the realistic nature of the social network, it seems that the Minimum Weighted Positive Influence Dominating Set (MWPIDS) problem can support the mentioned characteristics because it consider the weight of the graph. In this paper, a learning automaton based algorithm is proposed to reveal MWPIDS in the social network graphs. In the proposed algorithm, each vertex of the social network graph is equipped with a learning automaton that determines the beeing candidate or non-candidate of the corresponding vertex to be in WPIDS or not. Owing to adaptive decision making characteristics of learning automata, the proposed algorithm significantly reduces the number of candidate solution. The proposed algorithm, based on learning automata, iteratively decreases the weight of the obtained positive influence dominating. In order to evaluate the proposed algorithm, several experiments have been conducted on real social network datasets which compared to the state-of-the art methods. Experimental results show the superiority of the proposed algorithm over the previous algorithms.

Keywords—Learning Automata; Social Network Analysis; Dominating Set; Weighted Positive Influence Dominating Set

I. INTRODUCTION

Due to the popularity of online social networks such as *Facebook*, *Myspace* and *Linkdin* provide companies new setting for enabling large-scale marking through the powerful word-of-mouth effect. Recently, extensive researches around social network analysis have been investigated. They have shown that online users tend to share a new information or trust to a news which is gained from their close social groups such as friends and acquaintances instead of trust to comments of ordinary people or news from public media.

Many researchers on social network analysis have been focused on significant characteristic of social networks. In most of the studies, a social network can usually be represented as a network with a set of nodes corresponds to online users and edges representing a relationship between users. One of the significant characteristic of social networks is the viral marking that indicates the tendency of gaining largest influence on people through adopting the product. In [1] is shown that the viral marking is efficient than other marketing policies.

From the algorithmic aspect the term of viral marketing is transformed into the Influence Maximization (*IM*) that is known as an optimization problem. The main goal of the influence maximization in a social network is to find small k vertices (referred to as initial seeds) in the networks, such that under the certain diffusion model, the expected number of vertices influenced by *k* is maximized. In this paper, we propose a learning automata based algorithm for solving Minimum Weighted Positive Influence Dominating Set (MWPIDS) considering the characteristics in the social networks. On the other hand, the theory of learning automata has been shown its potential as an adaptive decision making unit in the dynamic and complex environment such as sensor networks[2], peer to peer networks[3] and social networks[4].

The rest of the paper is organized as follows: In section 2 reviews related work. In section 3 the theory of learning automata is introduced briefly. In section 4 the proposed learning automata is described. The performance of the proposed algorithm is evaluated through the simulation experiments in section 5. We make conclusion and future works in section 6.

II. RELATED WORKS

Clark et al. [5] proved that finding the *Minimum Dominating Set* and its variations are NP-hard and proved exponential time complexity. Several real practical applications of this problem are introduced in social networks. For example, in terms of influence maximization due to the policy of product adoption, one person can change the decision by accepting or refusing an offer. There are many researchs have been focused on Positive Influence Dominating Set (PIDS), however, there are only a little works done on Weighted Positive Influence Dominating Set. In the following, we will reviewed the outcomes of these studies in a nutshell: The milestone work for minimum positive influence dominating set problem is presented by *Wang et al*. [6]. They proposed an approximate greedy algorithm and proved that the problem is APX-hard. Moreover, the author's studied the PIDS(Positive Influence Dominating Set) problem under the deterministic linear threshold model, in which the influence from a pair of nodes is computed by total incoming influence from its neighbors exceeds a pre-determined threshold.

Recently, *Raei et al.* [7] proposed a new greedy algorithm which improves the time complexity of the primitive algorithm for PIDS. Although the greedy algorithm achieves many promising results, but in some cases it is limited to find proper approximation and produce unfavorable results. Zhang et al[8] studied PIDS in power-law graph and proved the greedy algorithm with constant factor. *Dineh et al.* [9] studied the PIDS from different points, and proposed the *Total Positive Influence Dominating Set* (TPIDS) instead of PIDS and considered ρ fraction of neighbors instead of half. In addition, they proved an approximation factor for TPIDS and showed that in power-law networks which follow power-law degree distribution, both PIDS and TPIDS admit constant factor approximation. *Basuchowdhuri et al.* [10] investigated a more generic problem which is similar to finding minimum dominating set, but it is different from the naive algorithm in terms of the number of hops needed to reach all nodes in the network and known as Minimum k-Hop Dominating Set. Intuitively, by increasing the parameter k corresponding to the number of hops, it increases the number of nodes which is triggered by the seeds. In [11] investigated partial positive influence dominating set and applied general threshold model for all nodes in networks. Threshold function is determined as a value of certain fraction of neighbors which are in active state.

A remarkable application of PIDS is maximizing the influence among participant in networks. Given a social network, the problem of influence maximization refers to select a set of nodes such that the spread of influence based on a given model is maximized. In order to modeling the micro behaviors of the individuals when they are exposed to specific information in networks, spreading models have been proposed. Two well-known models including independent cascading model (IC) and linear threshold model (LT) are presented in [12] . Based on spreading models, *Kempe et al.* [13] investigated the influence maximization problem in a social network for finding a set of initial nodes that the spread of information is maximized. *Wei Chen et al*. [14] investigate the IM, and modified the algorithms to apply in the context of large-scale data mining in real social networks. *Chen et al.* [15] proposed a heuristic based algorithm for influence maximization in arborescence which is called (MIA) to change the scalability of the algorithm using the maximum influence path of each pair of nodes. *Kim et al.* [16] proposed scalable and parallelizable influence approximation algorithm by calculating independent paths in networks. *Yang et al.* [17] extended the influence maximization problem and proposed a coordinate descent method to solve the transmission cost problem. *Goyal et al*. [18]used the simple path between neighbor nodes to estimate the influence spread in networks.

Due to the dynamic and complex nature of social networks, all works which were reviewed in this paper suffer from two major challenges that described as below.

- Positive and negative interactions through communication should be fully captured. Note that varying the behavior of the users reflected by positive and negative interactions affect the decisions of the algorithm which analyses the social network.
- Since the properties of users are different from each other, they reflect different levels of influence during the communication which IM ignores them.

None of existing algorithms reported for PIDS and its variations considers the mention characteristics. Our main contributions are listed as follows:

- We use a realistic model which is superior to model ude by the prior Influence Maximization algorithm.
- We design an efficient learning automata based algorithm to solve WPIDS in social networks. Our proposed algorithm is the first learning based algorithm for solving the WPIDS
- We experimentally proved that the obtained results are comparable and competitive with existing state-of-the art algorithms.

III. STATEMENT OF THE PROBLEM

The social networks can be represented by a graph $G = (V, E, W)$ where $V = \{V_1, V_2, ..., V_n\}$ is a set of nodes,

$E \subseteq V \times V$ is a set of edge and W indicates the weight function such that $F : v \rightarrow w_{v_i}$. Minimum Weighted Positive Influence Dominating Set (MWPIDS) is a subset of $S \subset V$ such that each node $v_i \in V$ is dominated by at least $\left\lceil \frac{n_i}{2} \right\rceil$ nodes in S with minimum weight where n_i is the degree of node v_i.

The topology of the underlay learning automata network can be represent by a graph $G^{'} = (V^{'}, E^{'})$ in which $V^{'} = \{LA_1, LA_2, ..., LA_n\}$ is a set of learning automata in underlying network and $E^{'} = V^{'} \times V^{'}$ is a set of links connecting the $LA^{'}s$ in the underlay network. In social networks, each node V_i is mapped to an LA_i in learning automata network based on one to one function $H : V \rightarrow V^{'}$. A primary goal of MWPIDS is to learn the select optimal action among a list of candidate ones in which is guided toward the optimal solution. Fig 1 demonstrates a snapshot of the graph of the social network.

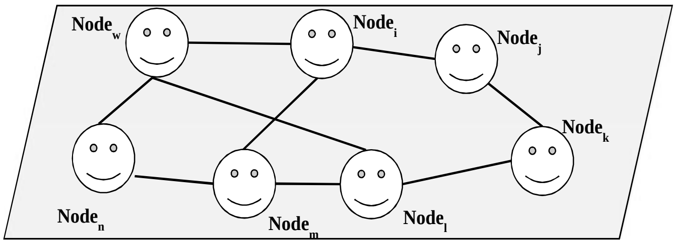


Fig. 1. A Snapshot of a graph of the social network

IV. LEARNING AUTOMATA

Learning automaton (LA) [12, 14] is an abstract model of finite state machine so that it can perform finite actions which randomly chosen and evaluated by a stochastic environment and response is given as a reward or penalty to LA. LA uses feedback of environment and select the next action. During this process, LA learns how to choose the best action from a set of allowed actions. Figure 2 illustrates the relationship between the LA and the it's random environment.

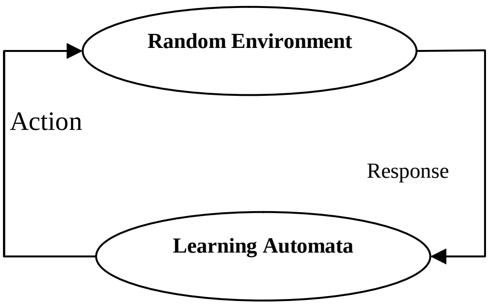


Fig.2. The relationship between the LA and its random environment

LA with variable structure can be described with quadruple $\{\sim, \ , P, T\}$ where $\sim = \{\sim_1, ..., \sim_r\}$ denotes the finite set of actions and $\ = \{\ _1, ..., \ _m\}$ is the set of input values that can be taken by reinforcement signal and $P = \{p_1, ..., p_r\}$ is the probability vector of each action. $P(n+1) = T[\sim(n), \ (n), p(n)]$ is learning algorithm where is updated in each iteration. Equations (1) and (2) illustrate changes in probability vector according to performance evaluation of α in each step. Automaton updates its action probability set based on equation (1) for favorable responses as:

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \qquad (1)$$

$$p_j(n+1) = (1-a)\, p_j(n) \qquad \forall j, = j \neq i$$

And equation (2) for unfavorable ones:

$$p_i(n+1) = (1-a)\, p_i(n)$$

$$p_j(n+1) = \left(\frac{b}{r-1}\right) + (1-b)\, p_j(n) \qquad \forall j, = j \neq i \qquad (2)$$

Where $P(n)$ is the action probability vector at instant n. r is the number of actions that can be taken by the LA. The learning rates a and b denote the reward and penalty parameters and determine the amount of increases and decreases of the action probabilities, respectively. If a=b learning algorithm is called linear reward penalty (L_{R-P}), if b<<a given learning algorithm is called linear reward-ε penalty ($L_{R-\varepsilon p}$), and finally, if b=0 it is called linear reward inaction (L_{R-I}) [24-26].

V. THE PROPOSED ALGORITHM

In this section, a learning automata based algorithm for finding MWPIDS problem will be proposed. In this algorithm, each node of the social network is equipped with a learning automaton which consist of two actions “candidate” and “non-candidate”. The network of the learning automata resided in the nodes of the social network has been utilized to find a proper WPIDS. For the sake of more clarification about the relationship between learning automata and the graph of the WPIDS Figure 2 is given. The detailed descriptions of the proposed algorithm are given in the rest of this section.

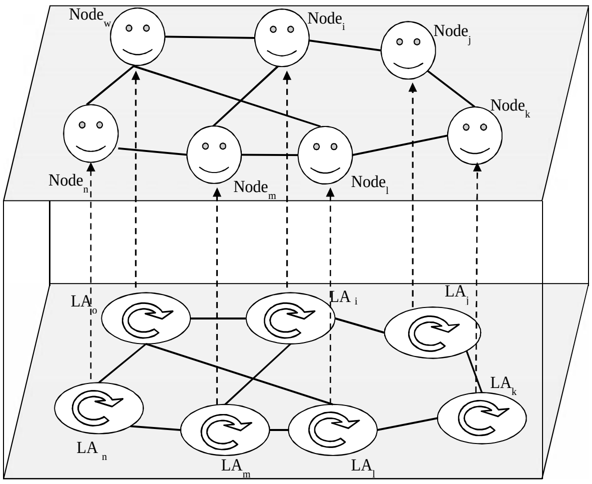


Fig2. A snapshot of mapping the network of the learning automata

The proposed algorithm consists of four steps: 1. Initialization, 2. Activation and action selection, 3. Verification and calculating reinforcement signal and, 4. Updating probability vector. These steps are described as follows: In the initialization, each node in the networks is equipped with learning automata that has two actions "candidate" and "non-candidate" and initially are disabled. At first all learning automata are activated and choosing its action randomly. Because of probabilities of all action in first iteration are equal, hence each action is selected with equal probability. After an action selection step is done the verification and calculating reinforcement signal is stars. In this step, the set of learning automata which selects the "candidate" actions are checked in terms of correctness, which means that the obtained set is WPIDS or not. Then their weights are calculated. If the weight of WPIDS is less than the former WPIDS then the selected action corresponds to current learning automata which is presented in the set are rewarded else the selects action are penalized. This process is continued until the weight of MPIDS is equal to pre-defined threshold. Figure 3 denotes the flowchart of the proposed algorithm.

Note that, at first, learning automata chooses its action with equal probability and will gradually converge to an appropriate value during iterations of the learning process. One of the most important parts in all learning algorithms is how the algorithm is terminated. In this way, there are various solutions are suggested. One of the common way is using a limitation on the number of iterations. If the number of iterations exceeds from a predefined threshold, then the algorithm stops. Another approach is computing the mean of the probability vector. After some iterations, if the mean of the probability vector is fixed then the algorithm stops. The mean of probabilities is expressed in equation (3)

$$Mean - probability = \frac{\sum_{i=1}^{n_{solution}} Max(P_i)}{|n_{solution}|} \tag{3}$$

Where $Max(P_i)$ corresponds to the maximum probability of learning Automata i and $n_{solution}$ demonstrated as the number of candidate LA which is present in solution. The general structure of proposed algorithm is shown in the following diagram as Figure 3. Thus LA learns how to choose the best action with for WPIDS in successive iterations. This process will be continued until the algorithm reaches to a near optimal solution. In the next section the experiment simulations are demonstrated in order to evaluate the proposed algorithm.

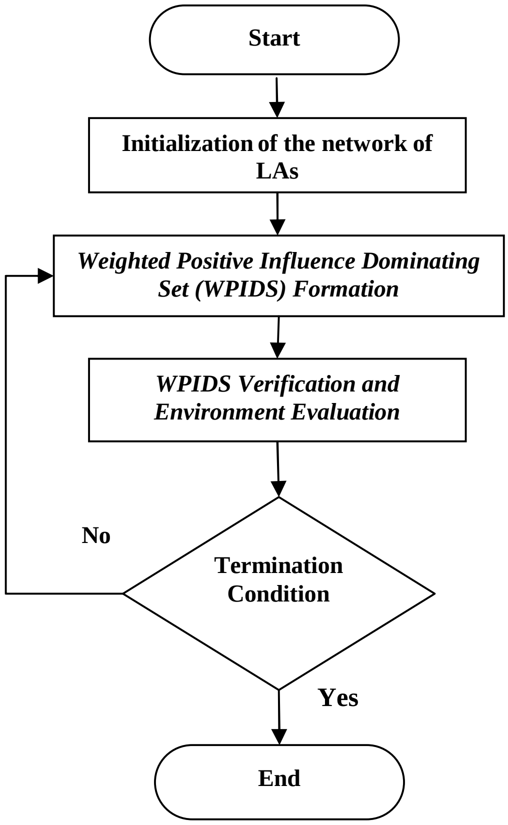


Fig3. Flow Chart of Proposed Algorithm based on Learning Automata for WPIDS.

VI. SIMULATION RESULTS

In order to evaluate the proposed algorithm, DIMACS [19] benchmark is used in experiments. DIMACS is a well-known benchmark. This benchmark contains different networks which are described in Table 1. For the proposed algorithm, all experiments have been implemented using L_{R-I} learning algorithm for the learning automata. Since the proposed algorithm operate on networks with vertex weight and there is no dataset for the social networks with vertex weight, the benchmark is modified using method reported by *Pullan et al.* [20] which is described as follows. Based on this method, the weight corresponds to node v_i can be computed by (i mod 200)+1. This method was also used in[21, 22].

Table1. Description of the datasets used for experiments.

Networks	*#nodes*	*#Edges*	D_{max}	D_{avg}
Brock200	*200*	*10k*	*114*	*98*
C250	*250*	*28k*	*236*	*223*
C500	*500*	*112k*	*468*	*449*
Dsjc500	*500*	*63k*	*286*	*250*

The results are reported in terms of the mean and standard deviation « ± ¢ with the average of $n \times 1000$ runs for the proposed algorithm. Note that n denotes the number of nodes in the network. The execution time of the proposed algorithm is reported in terms of minute.

The metrics used for the comparison are described as below.

- **Dynamic Threshold (DT)**: this metric computes the average weight of the PIDS.
- **Mean-Value:** This metric is computed using equation (3).
- **Number of activated nodes**: This metric computes the number of activated nodes of the network.

The design of the experiments is given as follows. The first experiment is conducted to study the impact of learning rate on the execution time and DT. The second experiment is designed to study the behavior of the proposed algorithm with respect to DT and Mean value during the process of finding the solution for WIPIDS in networks. The third experiment is conducted to study the impact of selecting influential nodes on the performance of the influence maximization algorithms with respect to the number of activated nodes.

Experiment 1

This experiment is carried out to study the impact of rate *a* on the performance of the proposed algorithm performance of the proposed algorithms. For this purpose, the *WPIDS* algorithm is evaluated for five different learning parameter {0.001, 0.01, 0.0.5, 0.1, 0.5}. As can be seen in Figure 5 and Figure 6, the dynamic threshold and execution time versus different learning rate are plotted respectively. The results show that the proposed Algorithm for which learning rate is small, works better than when the learning rate is large due to avoiding local minimum. But choosing a small learning rate may increase the learning time. For this problem, we have shown that experimentally the learning rate 0.05 can create a trade-off between execution time of the learning algorithm and the accuracy of the obtained solution is determined by the learning algorithm. Hence, for the rest of the experiment, we select 0.05 as the learning rate for the learning algorithm.

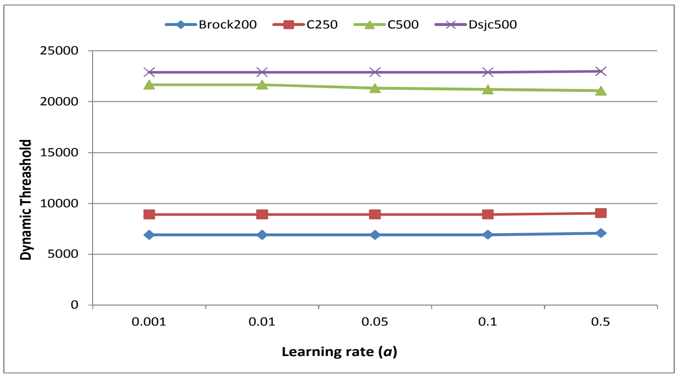


Fig 4.

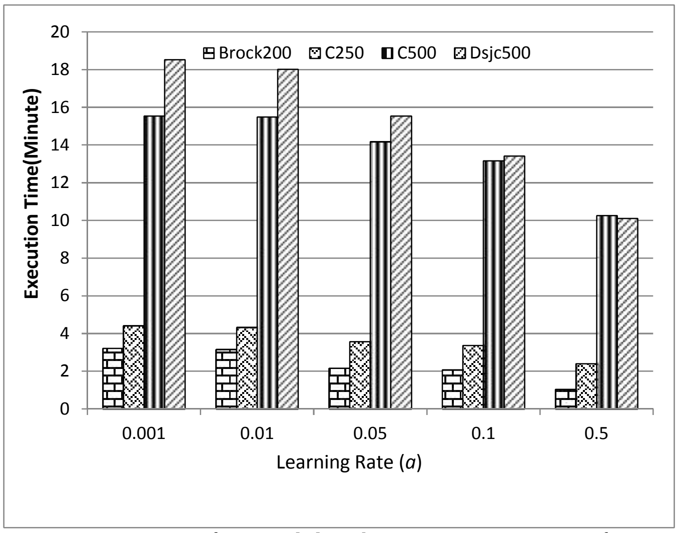
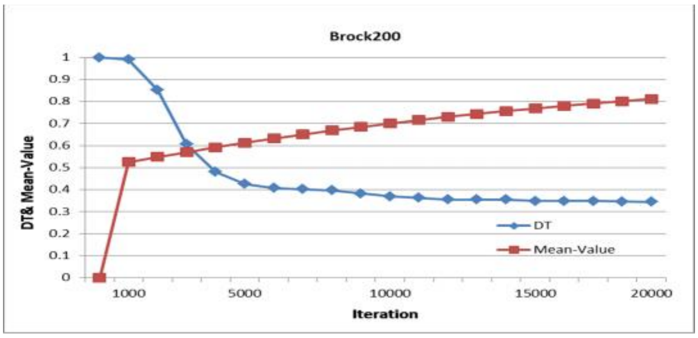


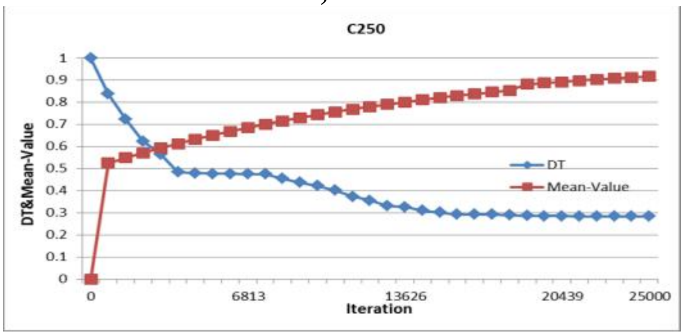
Fig. 5. Comparison of proposed algorithm in terms execution time for different learning rate.

Experiment 2

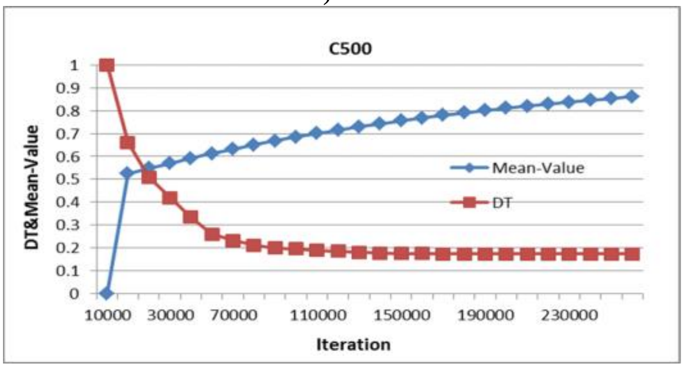
This experiment is investigated to study the behavior of the proposed algorithm during the process of finding the solution for WIPIDS in networks. In order to perform this experiment, we plot both *DT* and *Mean-Value* versus iterations. We note that *DT* is scaled between (0, 1) by dividing *DT* by the weight of maximum WPIDS for each network. The results of this experiment for different test graphs are given in Fig. **6.** As it is shown, *DT* gradually converges to the weight of minimum positive influence dominating set and at the same time *Mean-Value* converges to 1. This means that the algorithm gradually tends to find the WPIDS with minimum.



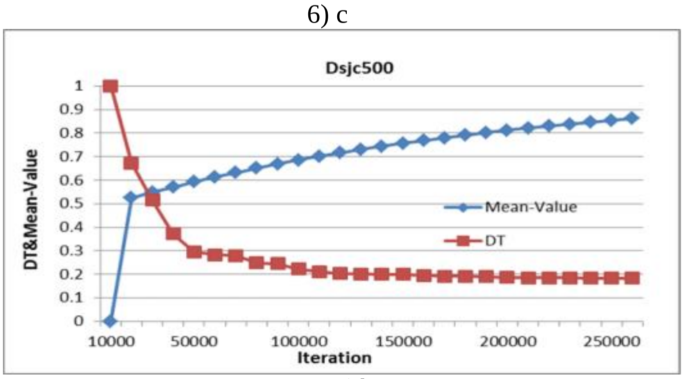
2) a



6)b



6) c

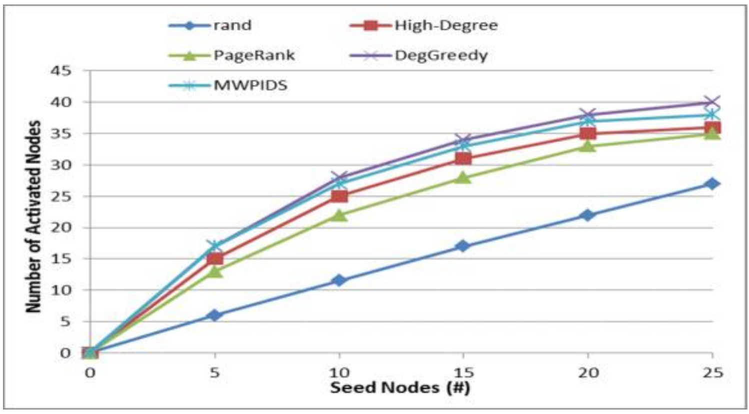


6)d

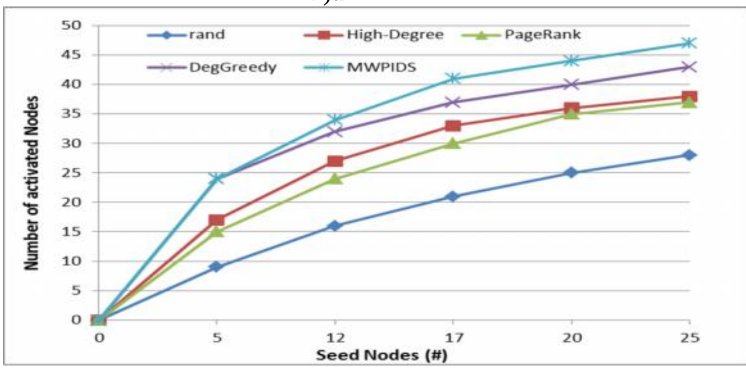
Fig 6. The plot of DT *and Mean-Value* versus iteration number for different Networks

Experiment 3

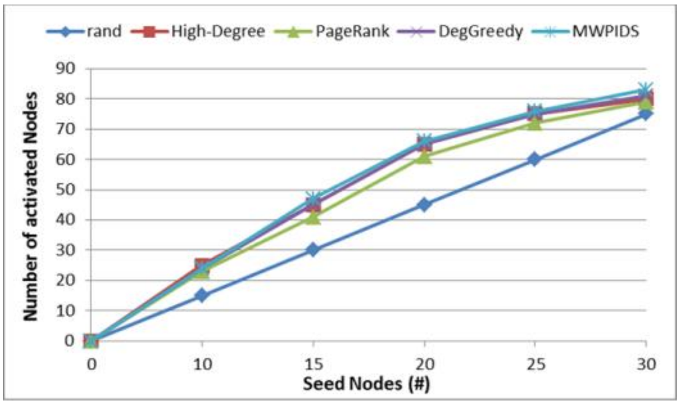
The goal of this experiment is to study the impact of selecting influential nodes on the performance of the influence maximization with respect to number of activated nodes. Based on these experiment our algorithm is compared with four well-known algorithm which is reported for influence maximization including: ,High-degree [12], PageRank[23], DegGreedy[24]and random generation algorithm on the all datasets. In High-degree algorithm k nodes is obtained based on the maximum degree. For PageRank algorithm, we have chosen the k highest ranked nodes as initial seed. The Page-rank algorithm computes and ranked nodes based on the significance. The DegGreedy algorithm try to maximize the influence based on node neighborhoods greedily and proved efficiency of the algorithm in terms of scalability and higher spread of influence. In random nodes are selected uniformly at random. For this experiment, the learning rates of the LA's are set to 0.05. In addition, for all networks, we assume that the threshold $\overline{} = 0.5$ based the Linear Threshold model. In addition we considered that all undirected datasets are double weight and directed for influence maximization, in which the weight node *i* is equal to $1/d_i$ where d_i is the degree of node *i*. Figure 7 shows number of activated nodes in comparison with initial seed size for all data sets.



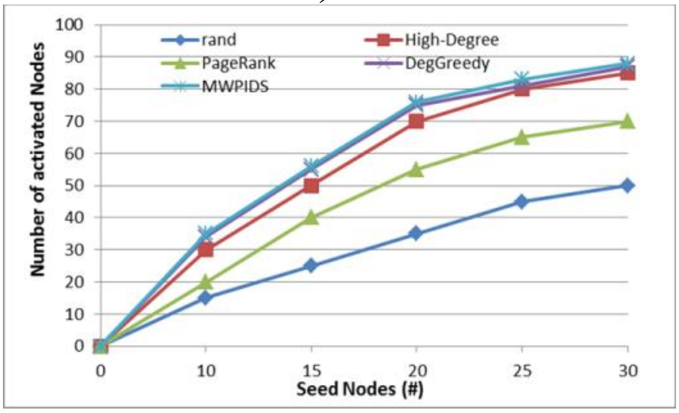
7)*a*



7)b



7)C



7)D

Figure 7: Comparing spread of influence using random, high-degree, PageRank, degGreedy and WPIS algorithms for the datasets

As can be seen from figure 7, it's compared spread of influence using random, high-degree, PageRank, DegGreedy and WPIDS algorithms for the all datasets. From the result, we may conclude that, DegGreedy algorithm outperforms than other heuristic algorithms, in addition high-degree centrality algorithm which relying solely on the structural properties of a network doesn't make sense for influence maximization. On the other hand, we need to also explicitly capture the dynamics of information which is captured by WPIDS nodes in the networks.

VII. Conclusion

In this paper, an algorithm for minimum weighted positive influence dominating set in social networks using learning automata was proposed. This algorithm is the first algorithm for the minimum weighted positive influence dominating set problem called MWPIDS in the literature. Based on the proposed algorithm, each node with the aim of a learning automaton is able to determine itself to the candidate or non-candidate for the MWPIDS. The proposed learning automata based algorithm, take the advantage of cooperation of learning automata with each other to find MWPIDS. In order to evaluate the performance of the proposed algorithm, a number of experiments have been conducted on popular networks. Experimental results showed that the proposed algorithm leading to promising results in terms of the number of activated nodes for influence maximization. Note that, due to the exponential complexity of WMPIDS problem, the algorithm is not supposed to reach to an optimal solution in reasonable time, but the results confirm that proposed algorithm produces better solutions than other well-known algorithms in terms of influence maximization.

VIII. References

[1] Fairbank, V., *A study into the effectiveness of viral marketing over the internet.* Retrieved April, 2008. **6**: p. 2013.

[2] Esnaashari, M. and M. Meybodi, *A cellular learning automata based clustering algorithm for wireless sensor networks.* Sensor Letters, 2008. **6**(5): p. 723-735.

[3] Saghiri, A.M. and M.R. Meybodi, *A Self-adaptive Algorithm for Topology Matching in Unstructured Peer-to-Peer Networks.* Journal of Network and Systems Management, 2016. **24**(2): p. 393-426.

[4] Rezvanian, A., M. Rahmati, and M.R. Meybodi, *Sampling from complex networks using distributed learning automata.* Physica A: Statistical Mechanics and its Applications, 2014. **396**: p. 224-234.

[5] Clark, B.N., C.J. Colbourn, and D.S. Johnson, *Unit disk graphs.* Discrete mathematics, 1990. **86**(1-3): p. 165-177.

[6] Wang, F., et al., *On positive influence dominating sets in social networks.* Theoretical Computer Science, 2011. **412**(3): p. 265-269.

[7] Raei, H., N. Yazdani, and M. Asadpour. *A new algorithm for positive influence dominating set in social networks*. in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012).* 2012. IEEE Computer Society.

[8] Zhang, W., et al., *Positive influence dominating sets in power-law graphs.* Social Network Analysis and Mining, 2012. **2**(1): p. 31-37.

[9] Dinh, T.N., et al., *On the approximability of positive influence dominating set in social networks.* Journal of Combinatorial Optimization, 2014. **27**(3): p. 487-503.

[10] Basuchowdhuri, P. and S. Majumder. *Finding influential nodes in social networks using minimum k-hop dominating set*. in *International Conference on Applied Algorithms*. 2014. Springer.

[11] Zhang, Z., et al. *Viral marketing with positive influence*. in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. 2017. IEEE.

[12] Kempe, D., J. Kleinberg, and É. Tardos. *Maximizing the spread of influence through a social network.* in *Proceedings of the ninth ACM*

SIGKDD international conference on Knowledge discovery and data mining. 2003. ACM.

[13] Kempe, D., J.M. Kleinberg, and É. Tardos, *Maximizing the Spread of Influence through a Social Network.* Theory of Computing, 2015. **11**(4): p. 105-147.

[14] Chen, W., Y. Wang, and S. Yang. *Efficient influence maximization in social networks*. in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2009. ACM.

[15] Chen, W., C. Wang, and Y. Wang. *Scalable influence maximization for prevalent viral marketing in large-scale social networks*. in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2010. ACM.

[16] Kim, J., S.-K. Kim, and H. Yu. *Scalable and parallelizable processing of influence maximization for large-scale social networks?* in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. 2013. IEEE.

[17] Yang, Y., et al. *Continuous influence maximization: What discounts should we offer to social network users?* in *Proceedings of the 2016 International Conference on Management of Data*. 2016. ACM.

[18] Goyal, A., W. Lu, and L.V. Lakshmanan. *Simpath: An efficient algorithm for influence maximization under the linear threshold model*. in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. 2011. IEEE.

[19] Tomita, E. and T. Seki, *An efficient branch-and-bound algorithm for finding a maximum clique.* DMTCS, 2003. **3**: p. 278-289.

[20] Pullan, W., *Approximating the maximum vertex/edge weighted clique using local search.* Journal of Heuristics, 2008. **14**(2): p. 117-134.

[21] Wu, Q. and J.-K. Hao, *A review on algorithms for maximum clique problems.* European Journal of Operational Research, 2015. **242**(3): p. 693-709.

[22] Gouveia, L. and P. Martins, *Solving the maximum edge-weight clique problem in sparse graphs with compact formulations.* EURO Journal on Computational Optimization, 2015. **3**(1): p. 1-30.

[23] Brin, S. and L. Page, *Reprint of: The anatomy of a large-scale hypertextual web search engine.* Computer networks, 2012. **56**(18): p. 3825-3833.

[24] Nandi, G., U. Sharma, and A. Das, *A Novel Hybrid Approach for Influence Maximization in Online Social Networks Based on Node Neighborhoods*, in *Advances in Electronics, Communication and Computing: ETAEERE-2016*, A. Kalam, S. Das, and K. Sharma, Editors. 2018, Springer Singapore: Singapore. p. 509-520.